CS 31006: Computer Networks – Application Layer

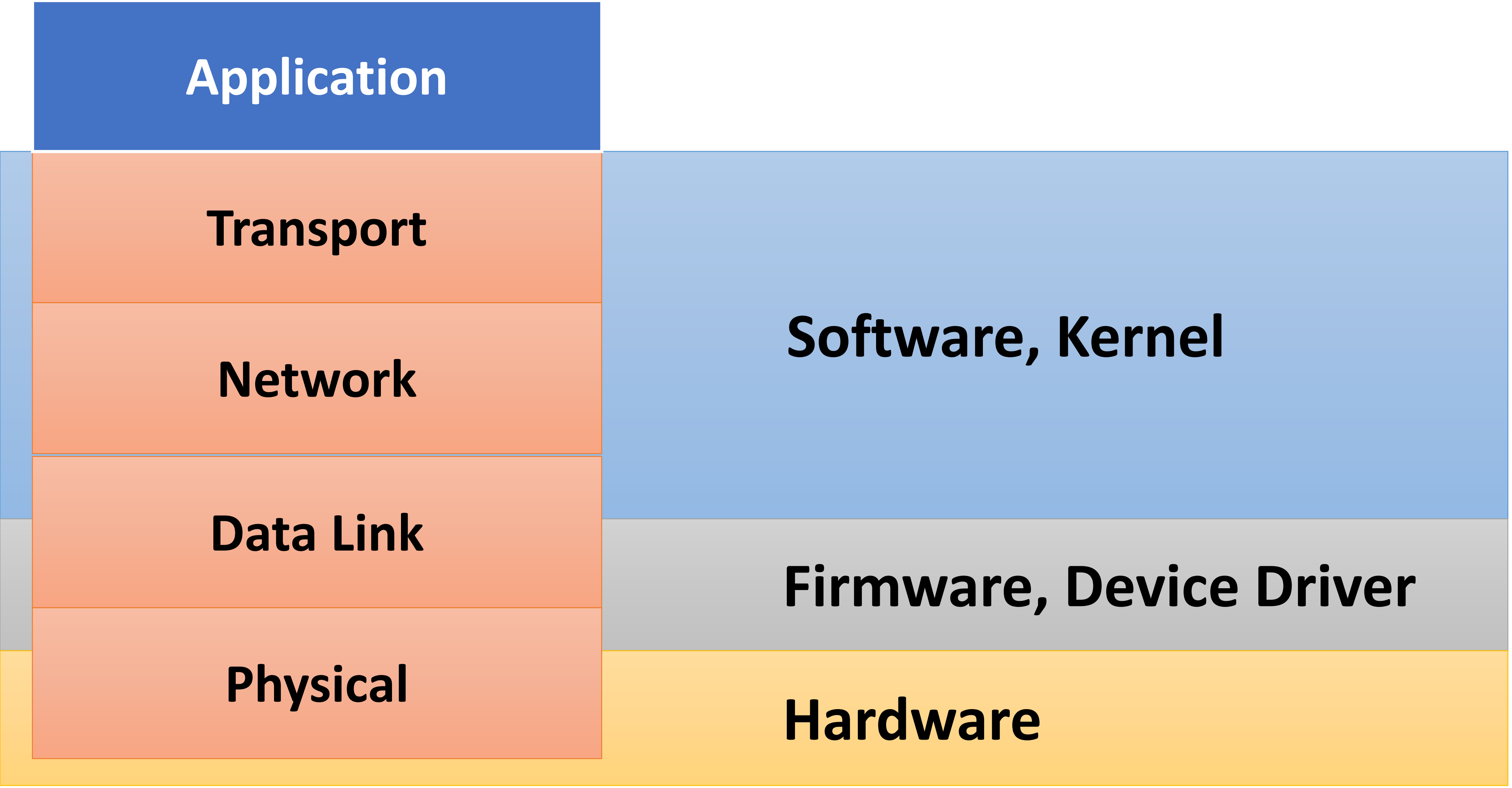
Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR

Rajat Subhra Chakraborty
rschakraborty@cse.iitkgp.ac.in

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

Protocol Stack Implementation in a Host



Application

Transport

Network

Data Link

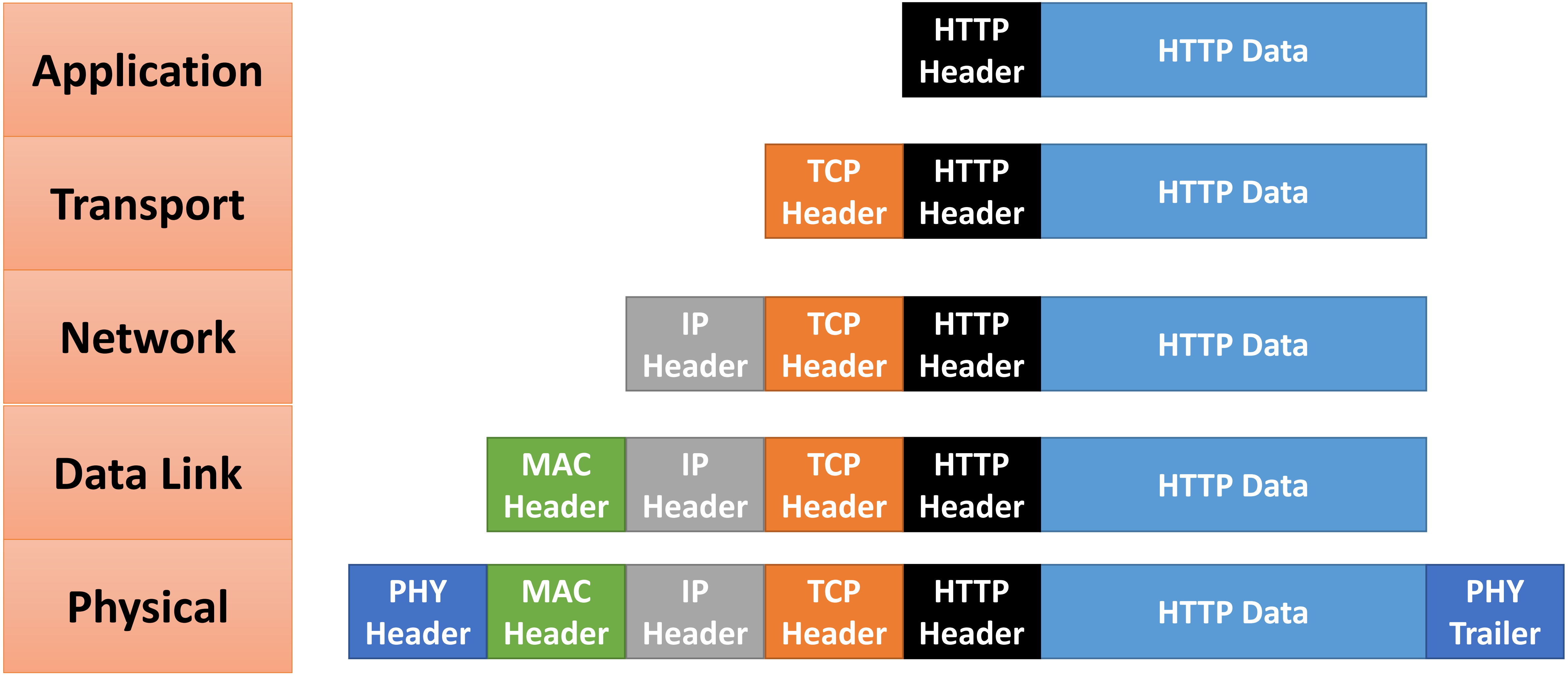
Physical

Software, Kernel

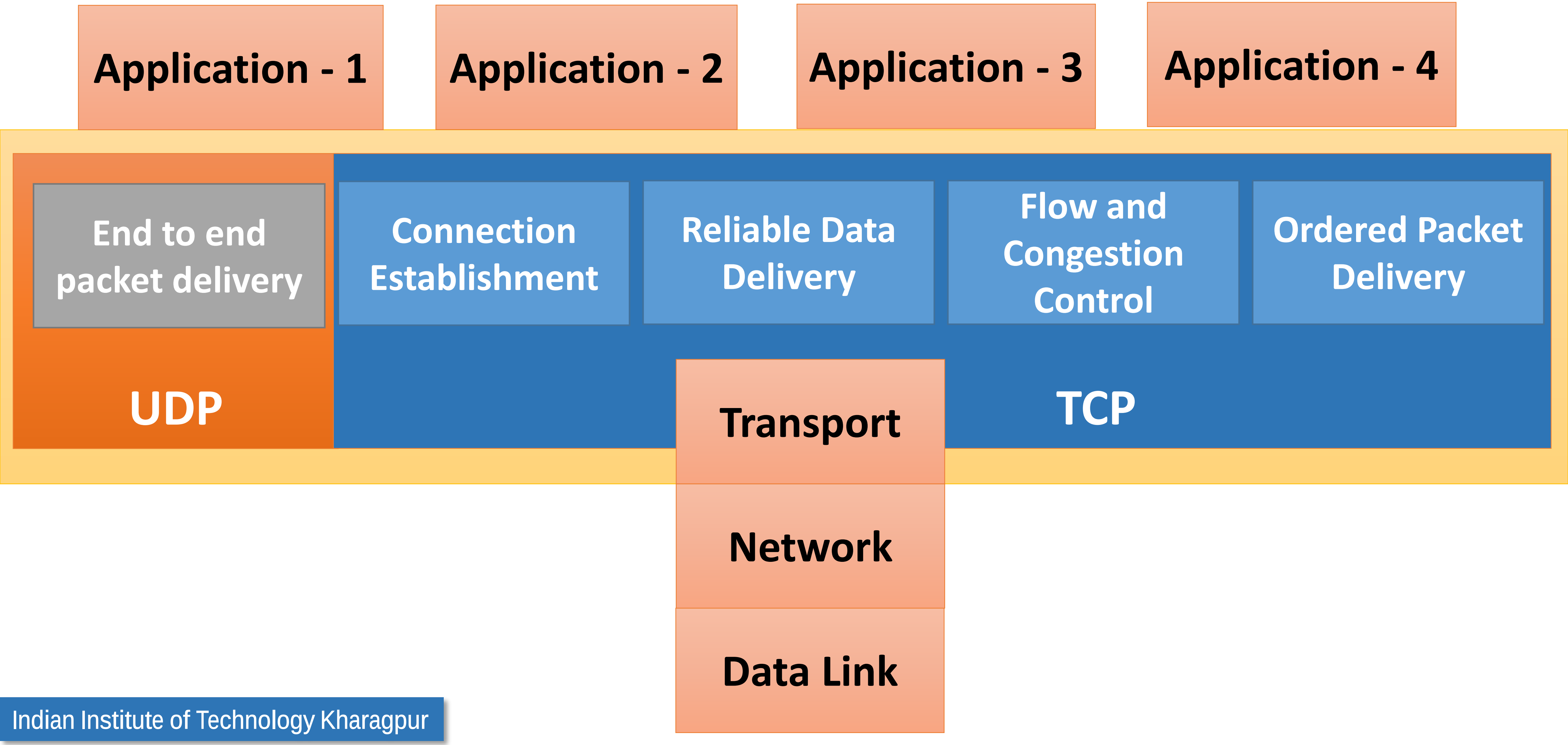
Firmware, Device Driver

Hardware

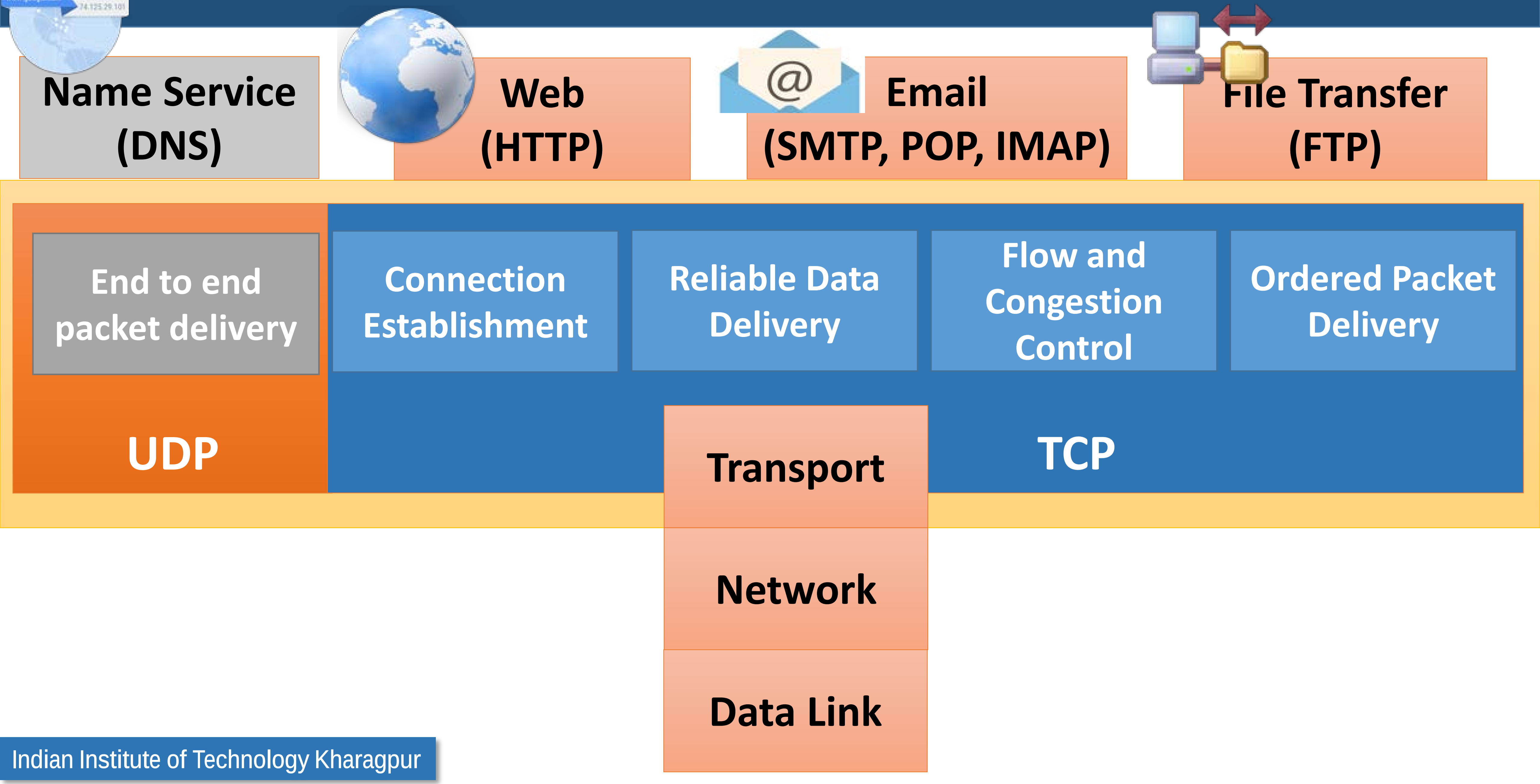
How Application Data Passes Through Different Layers



Application Layer Interfacing



Application Layer Interfacing



FTP

Name Service (DNS)

Web (HTTP)

Email (SMTP, POP, IMAP)

File Transfer (FTP)

End to end packet delivery

UDP

Connection Establishment

Reliable Data Delivery

Flow and Congestion Control

Ordered Packet Delivery

TCP

Transport

Network

Data Link

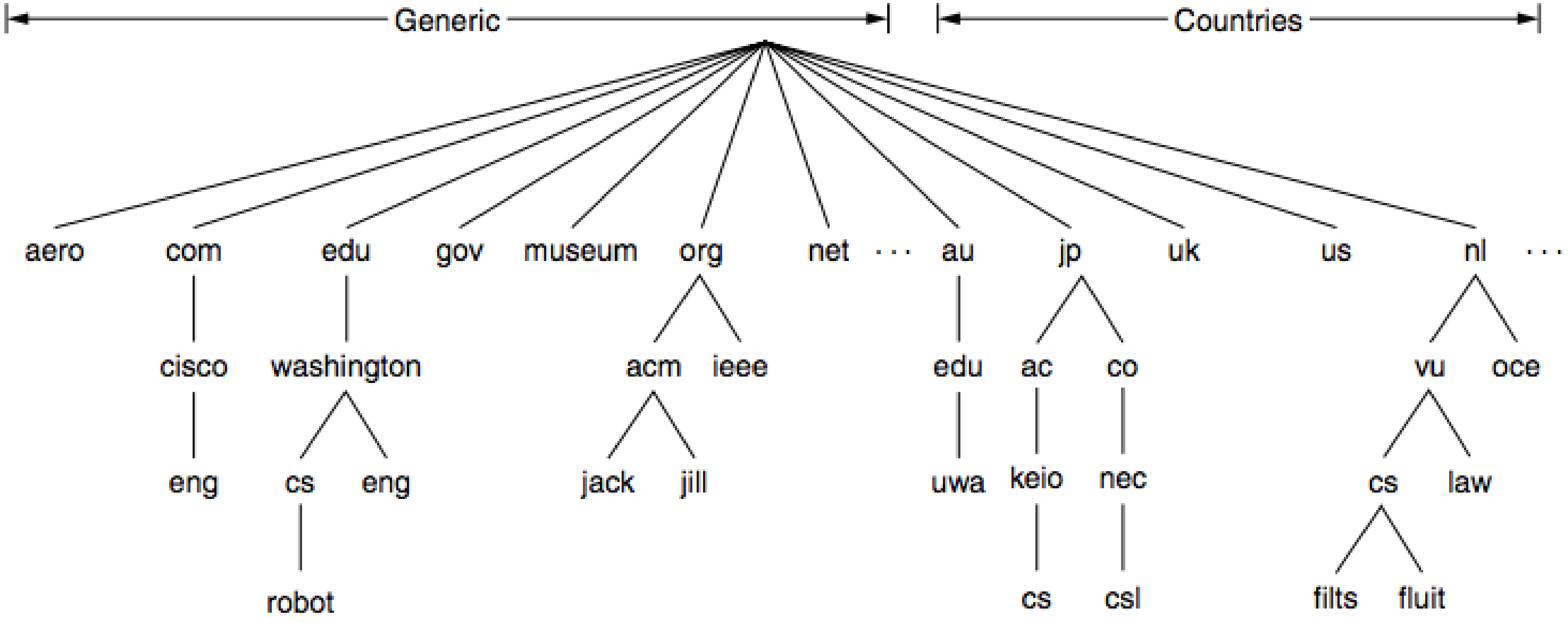
Domain Name System (DNS)

- Assign **Unique names** to an IP address (machine interface)

- ARPANET – a file *hosts.txt* listed all the computer names and their IP addresses.

- Resolve the host name conflicts over the Internet – a naming hierarchy needs to be managed
  - Rooted at an organization **Internet Corporation for Assigned Names and Numbers (ICANN)**

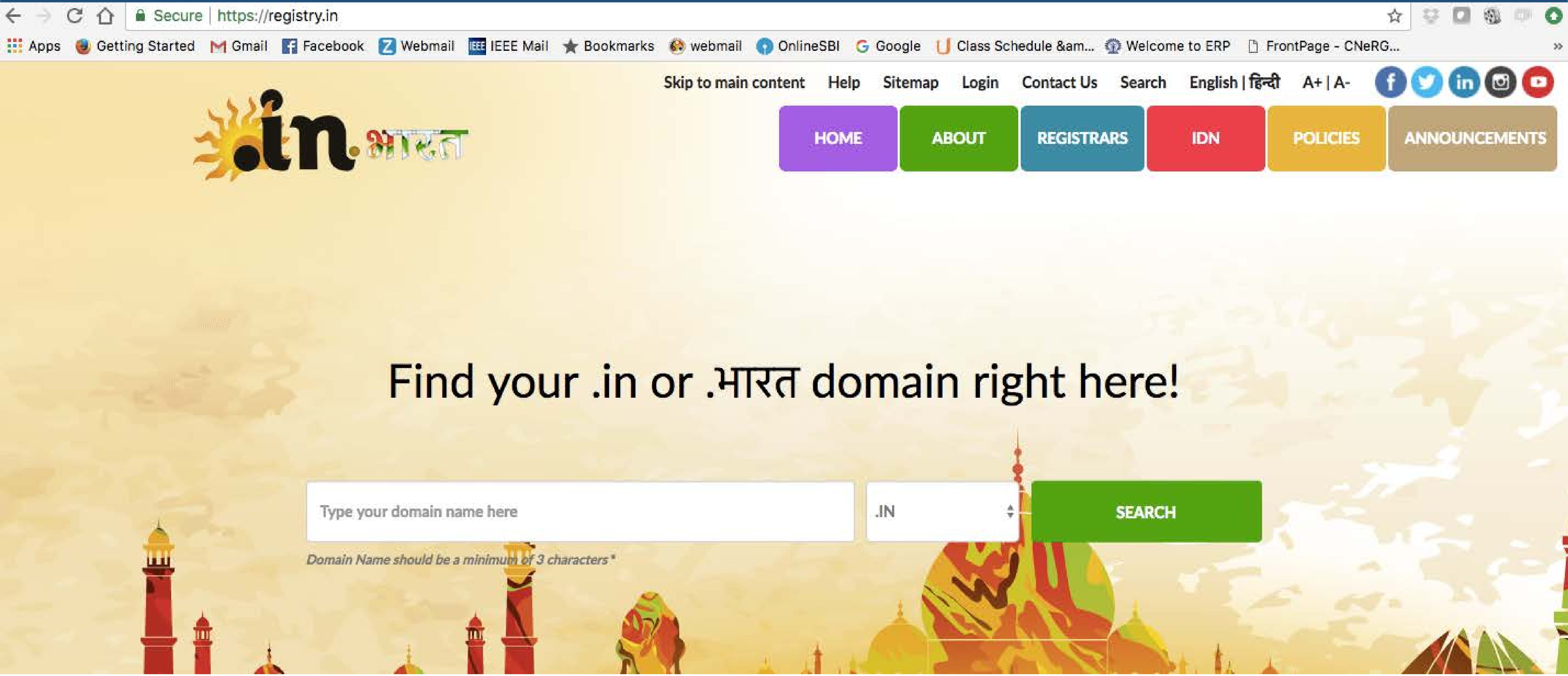
The DNS Name Space



Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

- The top level domains are run by **registrars** appointed by ICANN
- Name registrar for India (.in domain): **registry.in** (National Internet Exchange of India – **NIXI**)

registry.in

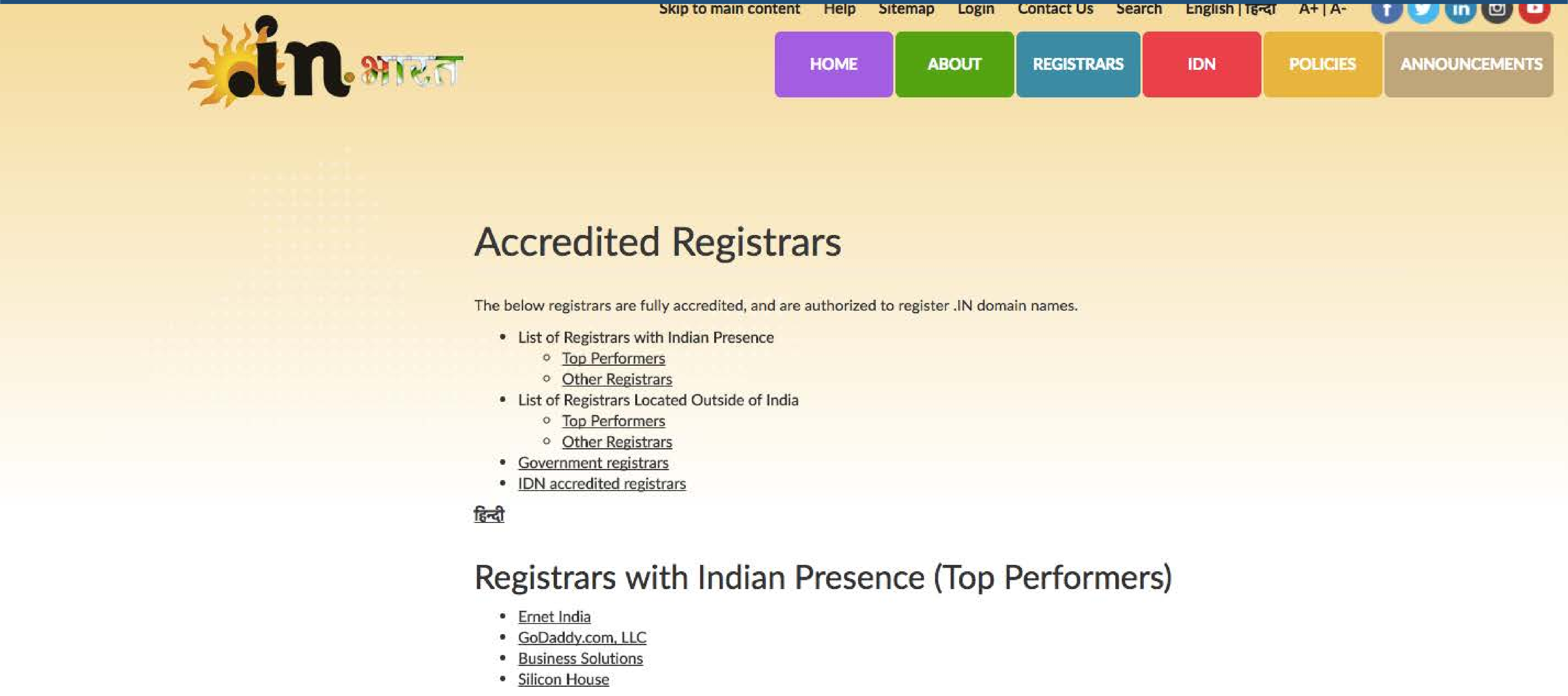


https://registry.in

Find your .in or .भारत domain right here!

*Domain Name should be a minimum of 3 characters**

Registrars Under registry.in



Accredited Registrars

The below registrars are fully accredited, and are authorized to register .IN domain names.

- List of Registrars with Indian Presence
  - Top Performers
  - Other Registrars
- List of Registrars Located Outside of India
  - Top Performers
  - Other Registrars
- Government registrars
- IDN accredited registrars

हिन्दी

Registrars with Indian Presence (Top Performers)

- Ernet India
- GoDaddy.com, LLC
- Business Solutions
- Silicon House

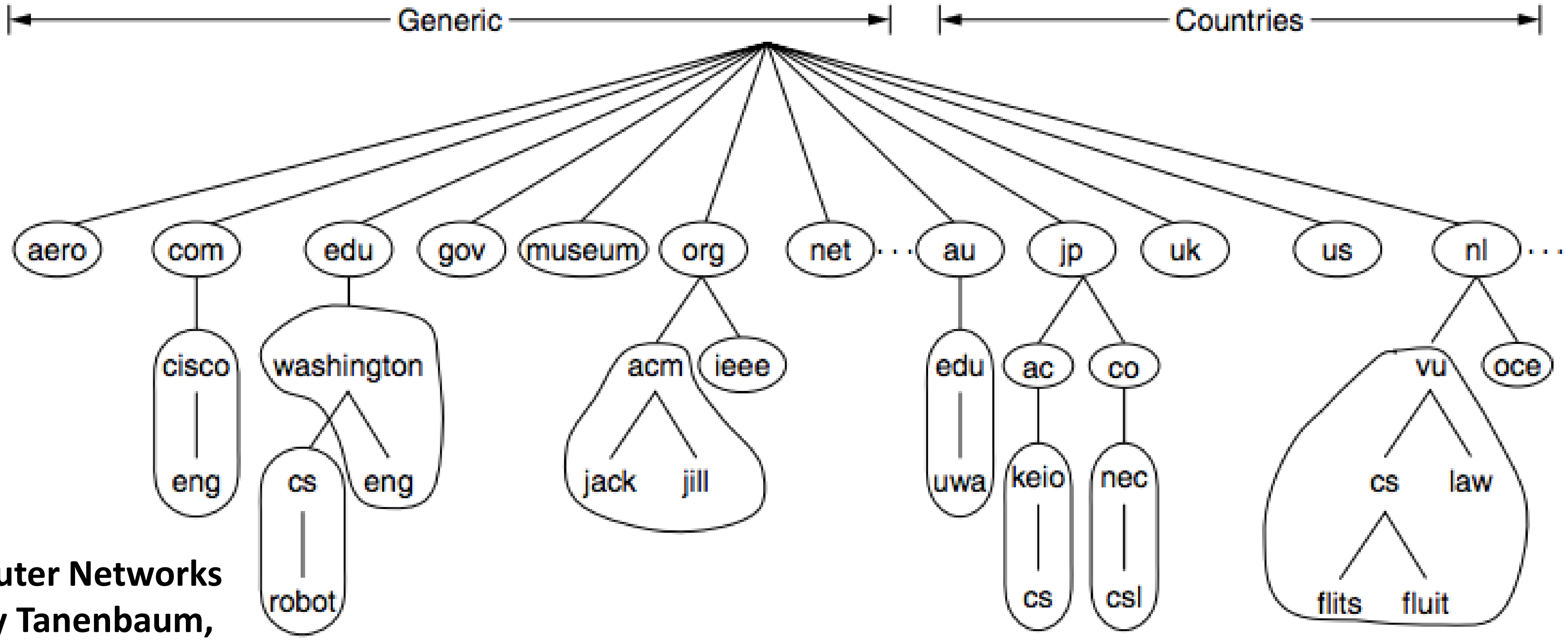
Elements of DNS (RFC 1034)

- **The Domain Name Space and Resource Records:** Specifications for a tree-structured namespace and data associated with names.

- **Name Servers:** Server programs which hold information about the domain tree's structure and set information
  - A particular name server has complete information about a subset of the domain space
  - Name servers know the parts of the domain tree for which they have complete information -- a name server is said to be an **AUTHORITY** for this parts of the namespace

Name Servers

- Divide registries into **non-overlapping** zones – every zone has a name server



Source: Computer Networks (5^{th} Edition) by Tanenbaum, Wetherell

Elements of DNS (RFC 1034)

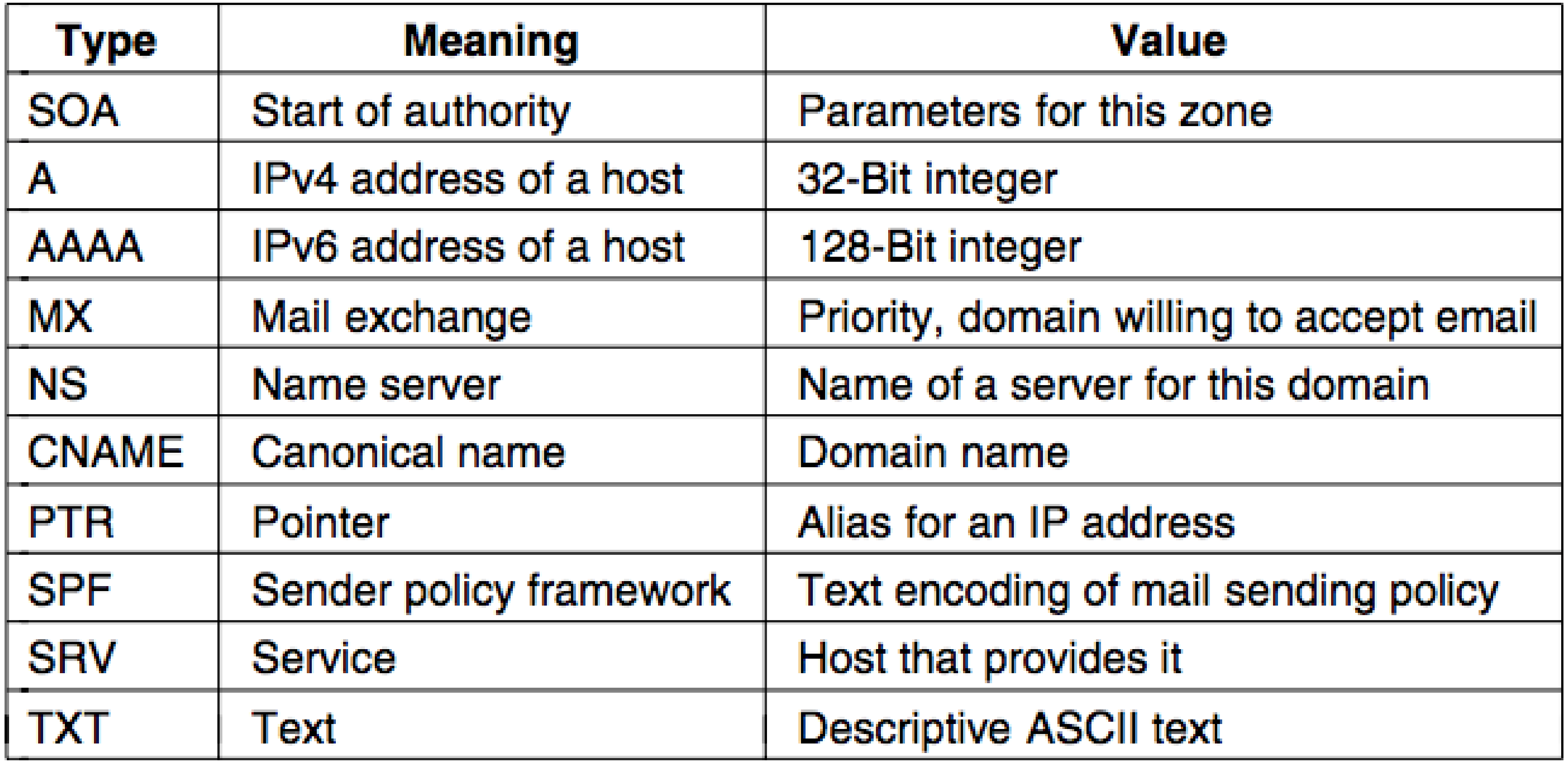
- **The Domain Name Space and Resource Records:** Specifications for a tree-structured namespace and data associated with names.

- **Name Servers:** Server programs which hold information about the domain tree's structure and set information
  - A particular name server has complete information about a subset of the domain space
  - Name servers know the parts of the domain tree for which they have complete information -- a name server is said to be an **AUTHORITY** for this parts of the namespace

- **Resolvers:** Program that extracts information from name servers in response to client requests

Domain Resource Records

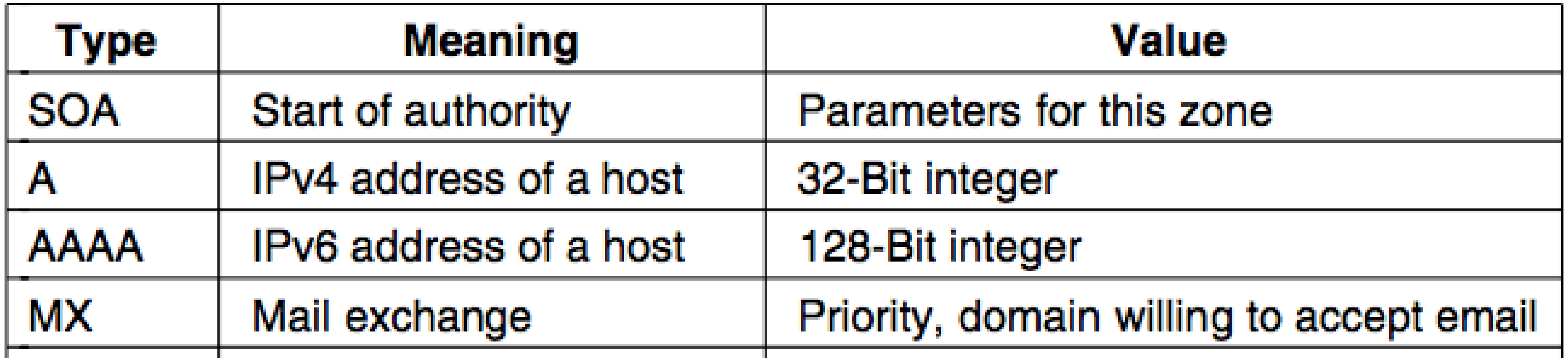
- Every domain has a set of **resource records** associated with it – DNS database



Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

Domain Resource Records

- **Domain_name**: Domain to which the record applies.
- **Time_to_live:** Time for which this record is active – volatile records may be assigned a small value
- **Class:** Normally IN – Internet resources
- **Type:** What type of record it is
- **Value:** Value of the record (IP address for A record type)



Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email

Domain Resource Records

- **A Type Records:**

```
cse.iitkgp.ac.in  86400    IN   A   203.110.245.250
```

- **CNAME Type Records:**

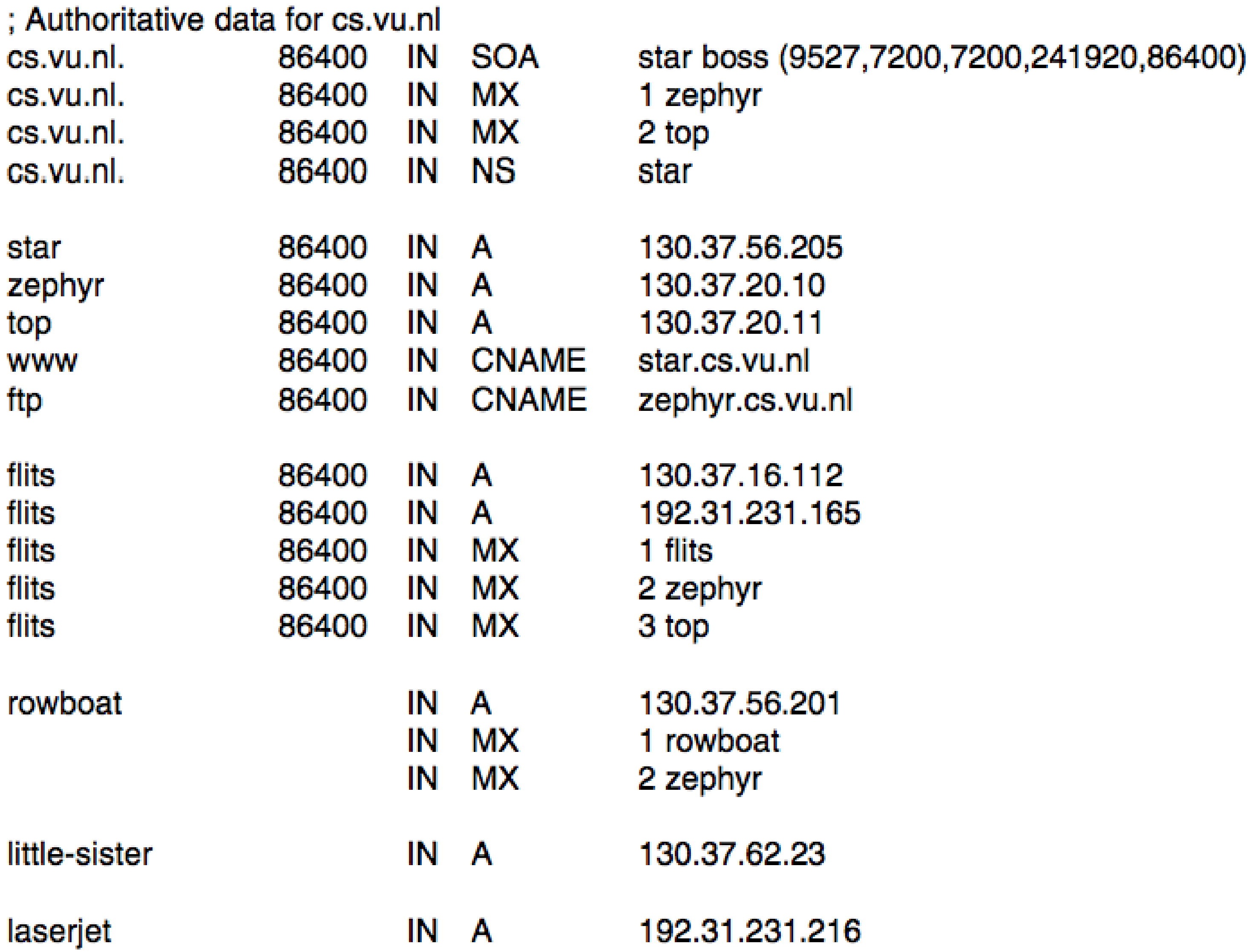
```
iitkgp.ac.in      86400    IN   CNAME www.iitkgp.ac.in
```

- **PTR Type Records:**

```
172.16.5.33       86400    IN   PTR www.iitkgp.ac.in
```

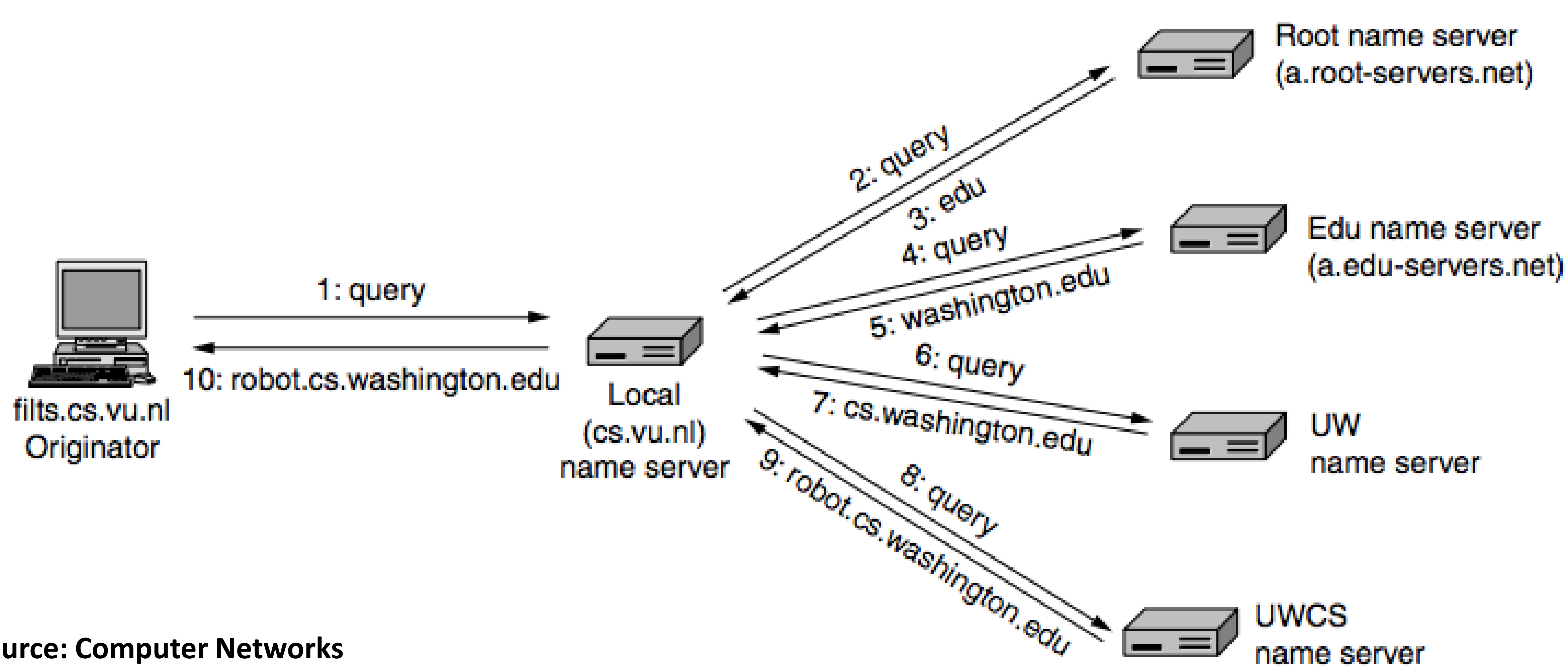
Sample DNS Database

Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell



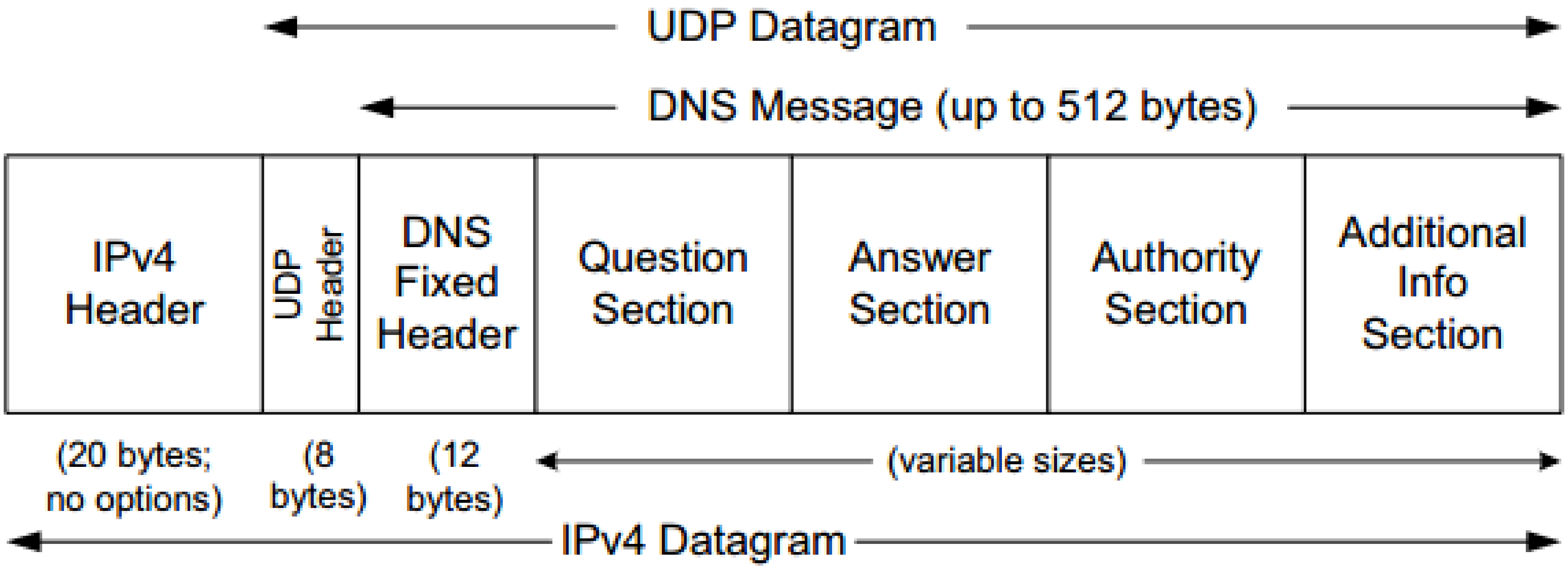
```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN   SOA     star boss (9527,7200,7200,241920,86400)
cs.vu.nl.       86400   IN   MX      1 zephyr
cs.vu.nl.       86400   IN   MX      2 top
cs.vu.nl.       86400   IN   NS      star

star            86400   IN   A       130.37.56.205
zephyr          86400   IN   A       130.37.20.10
top             86400   IN   A       130.37.20.11
www             86400   IN   CNAME   star.cs.vu.nl
ftp             86400   IN   CNAME   zephyr.cs.vu.nl

flits           86400   IN   A       130.37.16.112
flits           86400   IN   A       192.31.231.165
flits           86400   IN   MX      1 flits
flits           86400   IN   MX      2 zephyr
flits           86400   IN   MX      3 top

rowboat                 IN   A       130.37.56.201
                        IN   MX      1 rowboat
                        IN   MX      2 zephyr

little-sister           IN   A       130.37.62.23

laserjet                IN   A       192.31.231.216
```

Name Resolution – Looking Up for the Names

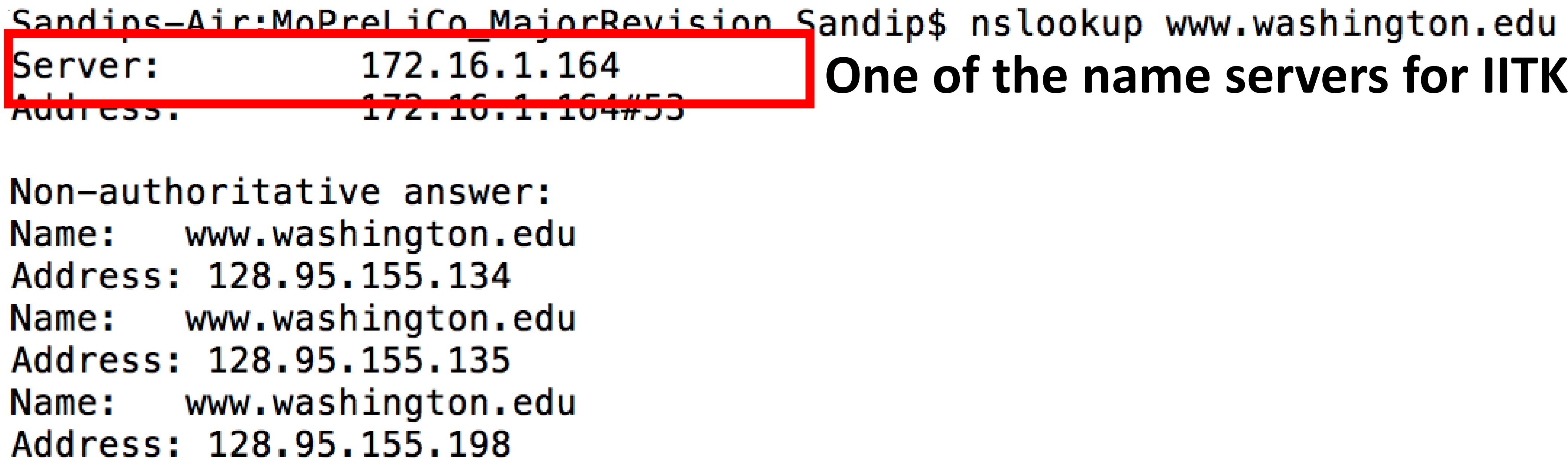


Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

DNS Packet Structure



Name Resolution (nslookup)



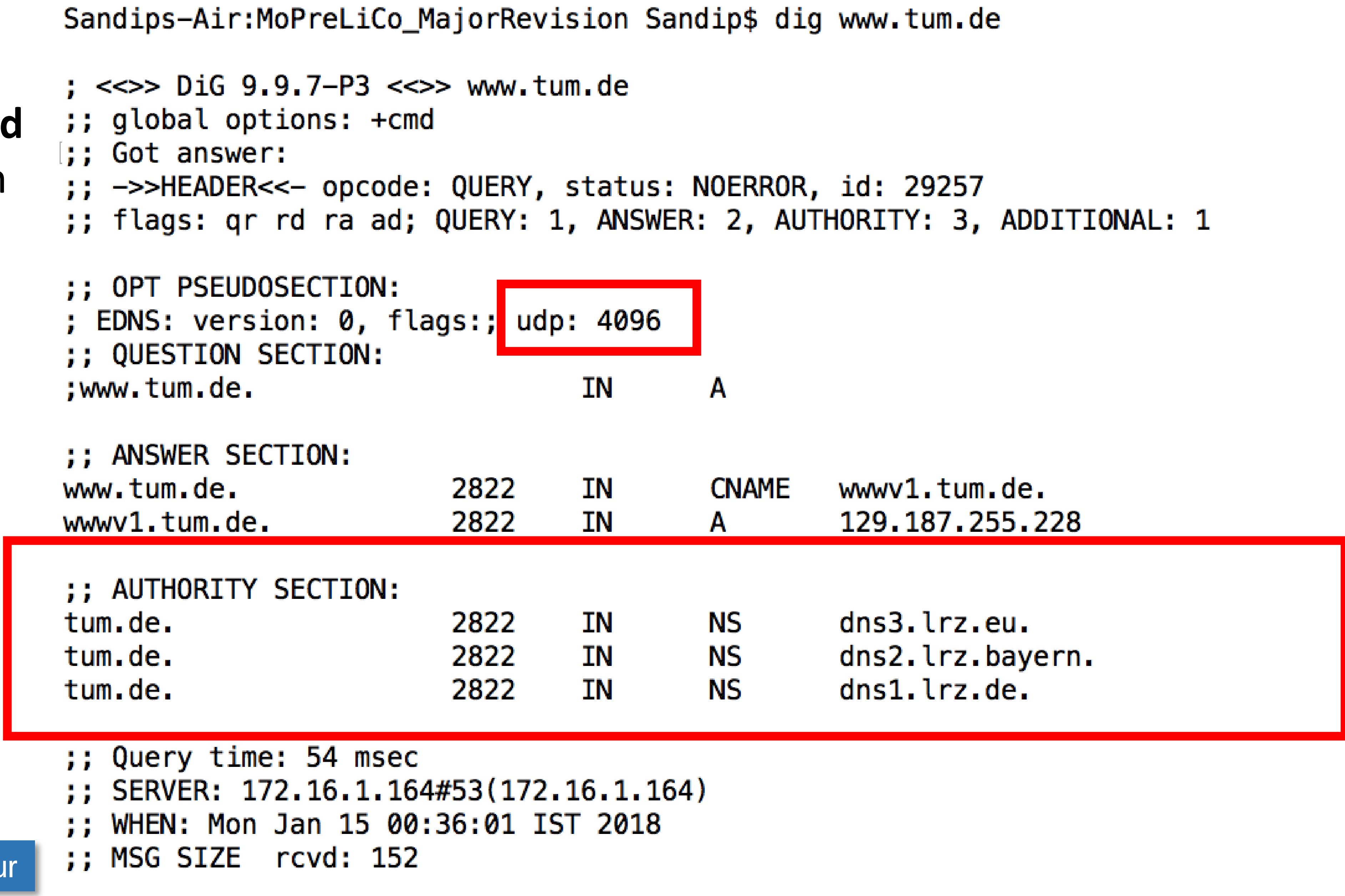
```
Sandips-Air:MoPreLiCo_MajorRevision Sandip$ nslookup www.washington.edu
Server:         172.16.1.164
Address:        172.16.1.164#53

Non-authoritative answer:
Name:   www.washington.edu
Address: 128.95.155.134
Name:   www.washington.edu
Address: 128.95.155.135
Name:   www.washington.edu
Address: 128.95.155.198
```

One of the name servers for IITKGP

Name Resolution (dig)

An **authoritative record** is one that comes from the authority that manages the record, and thus is always correct

UDP Message for 4096 bytes



```
Sandips-Air:MoPreLiCo_MajorRevision Sandip$ dig www.tum.de

; <<>> DiG 9.9.7-P3 <<>> www.tum.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29257
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.tum.de.                    IN      A

;; ANSWER SECTION:
www.tum.de.             2822    IN      CNAME   wwwv1.tum.de.
wwwv1.tum.de.           2822    IN      A       129.187.255.228

;; AUTHORITY SECTION:
tum.de.                 2822    IN      NS      dns3.lrz.eu.
tum.de.                 2822    IN      NS      dns2.lrz.bayern.
tum.de.                 2822    IN      NS      dns1.lrz.de.

;; Query time: 54 msec
;; SERVER: 172.16.1.164#53(172.16.1.164)
;; WHEN: Mon Jan 15 00:36:01 IST 2018
;; MSG SIZE  rcvd: 152
```

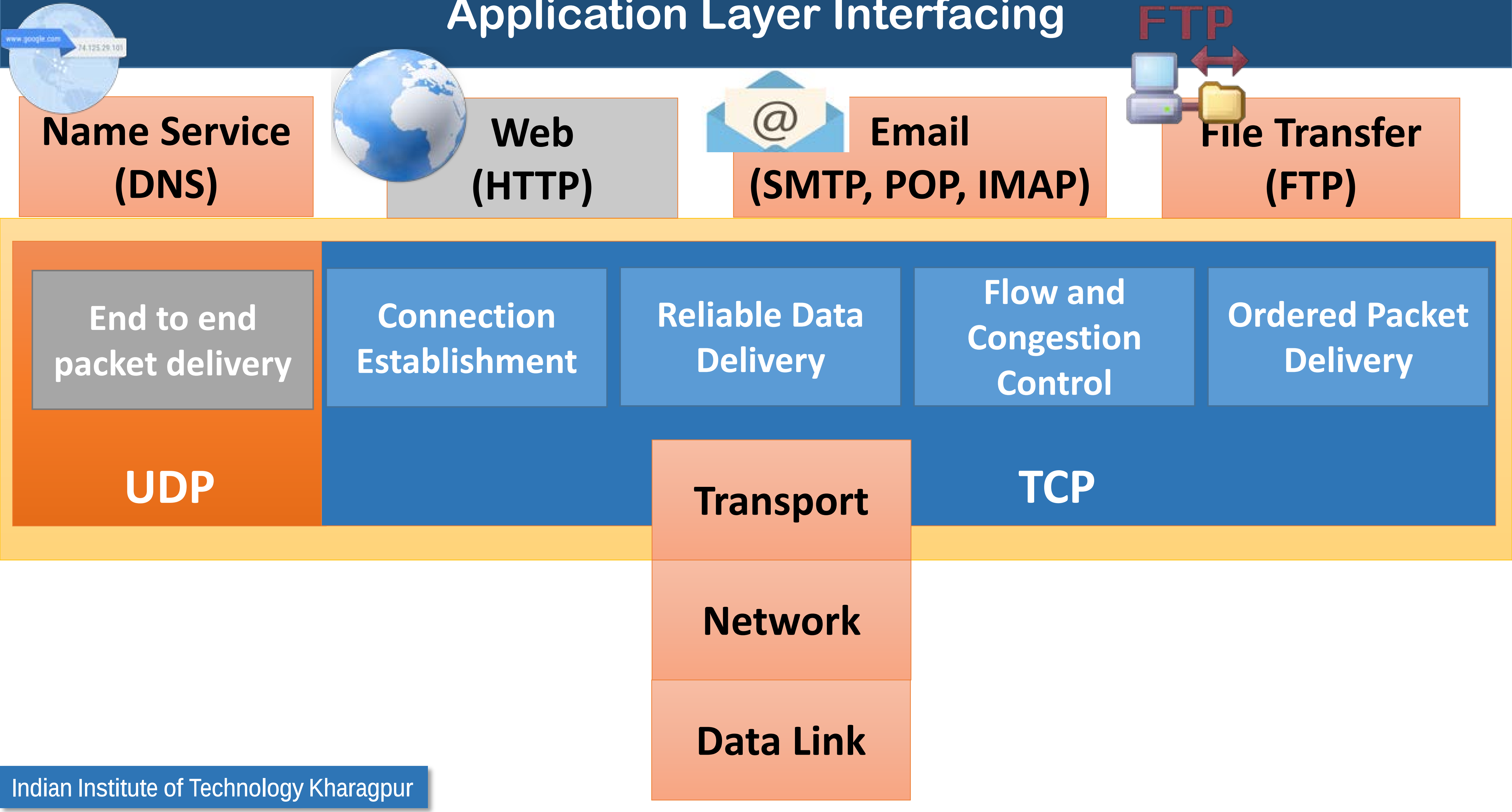
Why DNS Uses UDP

- UDP is much faster. TCP requires handshake time. DNS uses a cascading approach for name resolution. With TCP, for every message, a connection setup is required.

- DNS requests and responses are generally very small, and fits well within one UDP segment.

- **UDP is not reliable.** In DNS, reliability is ensured at the application layer. After timeout, the DNS client sends back the requests. After few consecutive timeouts (can be set at the client), the request is aborted with an error.

Application Layer Interfacing



Name Service (DNS)

Web (HTTP)

Email (SMTP, POP, IMAP)

File Transfer (FTP)

End to end packet delivery

UDP

Connection Establishment

Reliable Data Delivery

Flow and Congestion Control

Ordered Packet Delivery

TCP

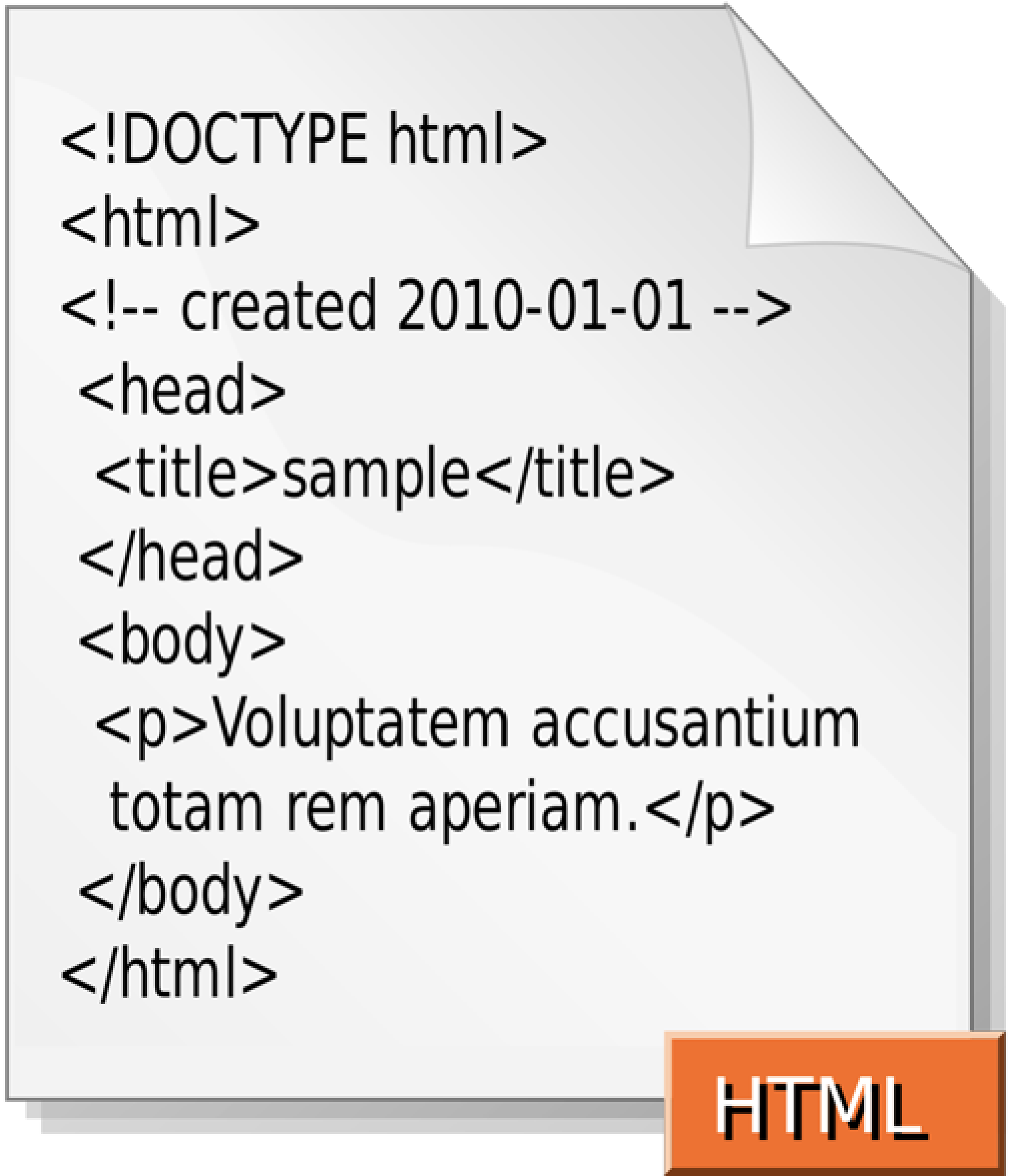
Transport

Network

Data Link

The Web – Hypertext Transfer Protocol (HTTP)

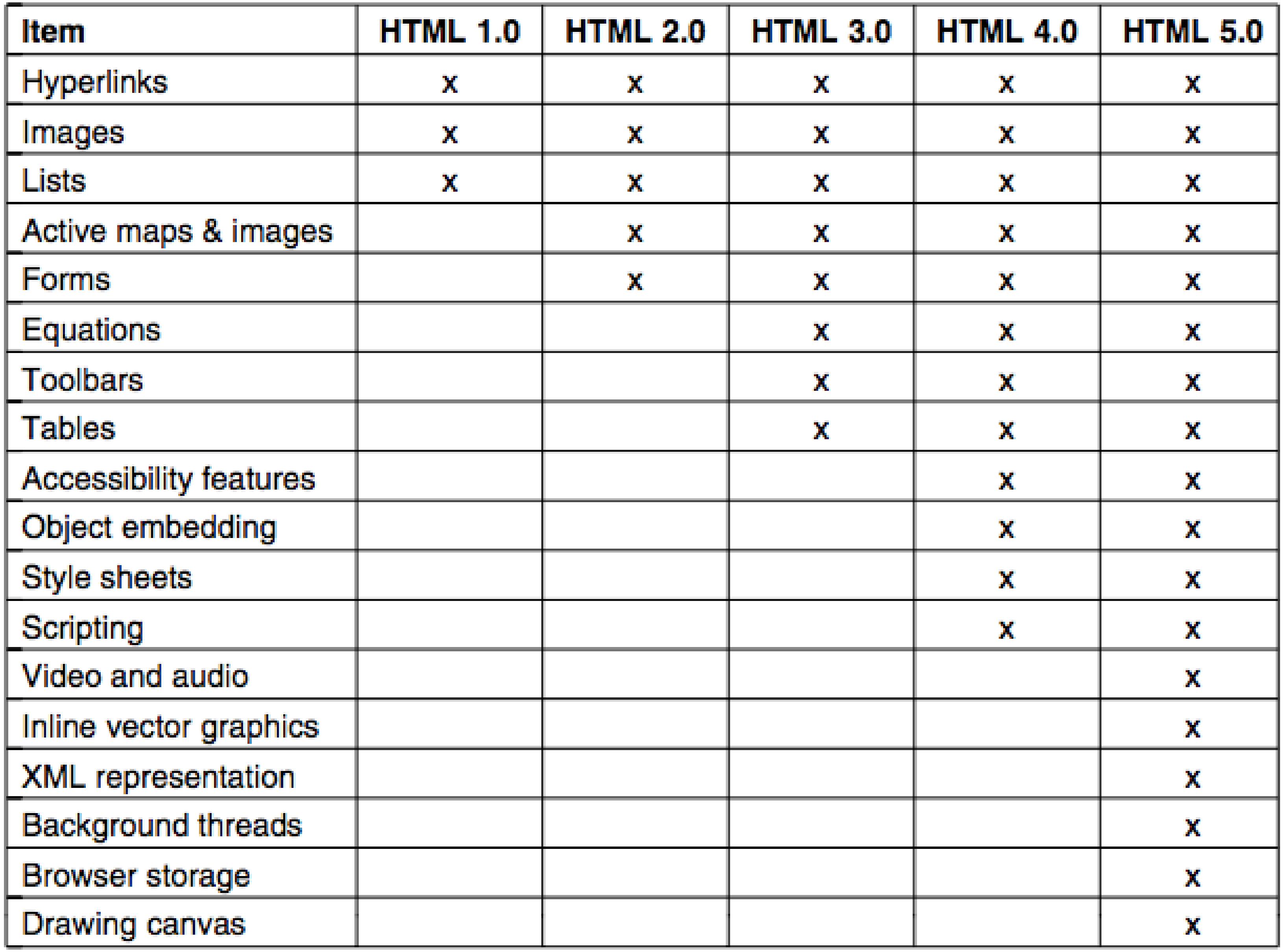
- **Hypertext -** A way to represent web content (text along with formatting)
- **Hypertext Markup Language (HTML)** - A scripting language to specify web data along with simple formatting (bold, italics, new line).
  - **A way to convert text based information to graphics based information**
- Today's era: Many graphics, scripts and other information are embedded inside HTML – CSS, JavaScript etc.



```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
 <head>
  <title>sample</title>
 </head>
 <body>
  <p>Voluptatem accusantium
   totam rem aperiam.</p>
 </body>
</html>
```

Differences between HTML Versions

Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell



Item	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
Hyperlinks	x	x	x	x	x
Images	x	x	x	x	x
Lists	x	x	x	x	x
Active maps & images		x	x	x	x
Forms		x	x	x	x
Equations			x	x	x
Toolbars			x	x	x
Tables			x	x	x
Accessibility features				x	x
Object embedding				x	x
Style sheets				x	x
Scripting				x	x
Video and audio					x
Inline vector graphics					x
XML representation					x
Background threads					x
Browser storage					x
Drawing canvas					x

A History of the Web

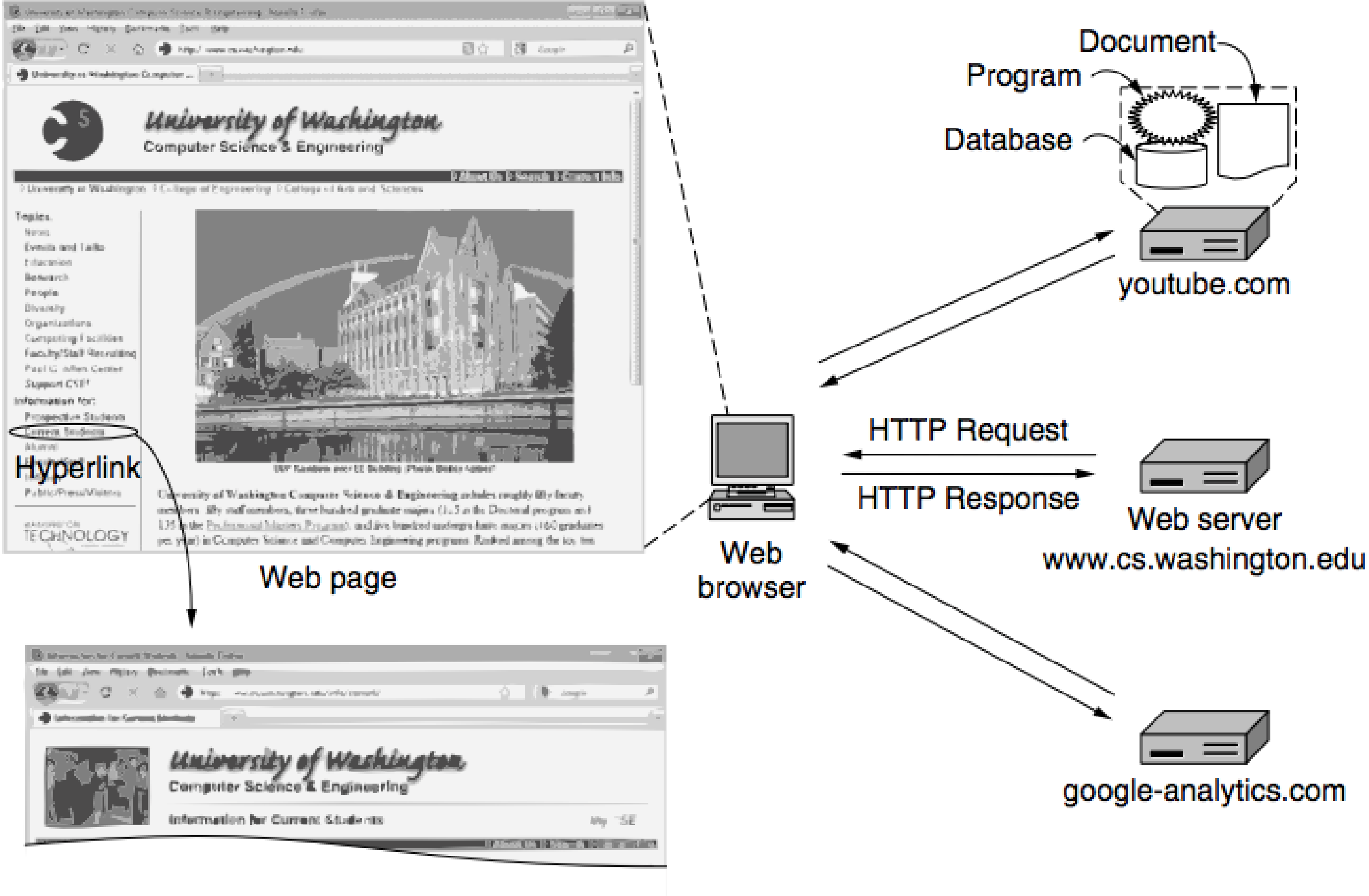
- 1989 (CERN – European Center for Nuclear Research) – help large teams to collaborate using a constantly changing collection of reports, blueprints, drawings, photos and other documents
  - The proposal came from Tim Berners-Lee
  - A public demonstration at Hypertext '91 conference

- 1993 – The first graphical browser (Mosaic) developed by Marc Andreessen, University of Illinois
  - Andreessen formed the company Netscape Communications Corp
  - Microsoft developed Internet Explorer – "browser war" between Internet Explorer and Netscape Navigator

- 1994 – CERN and MIT signed an agreement to setup **World Wide Web Consortium (W3C)**

The Web – Architectural Overview

Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell



HTTP – The Client Side

- Three questions to be answered for accessing a web page
  - What is the page called? (**courses.html**)
  - Where is the page located? (**cse.iitkgp.ac.in/~sandipc/**)
  - How can the page be accessed? (**http://**)

- **Uniform Resource Locator (URL):** Each page is assigned a URL that effectively serves the page's worldwide name

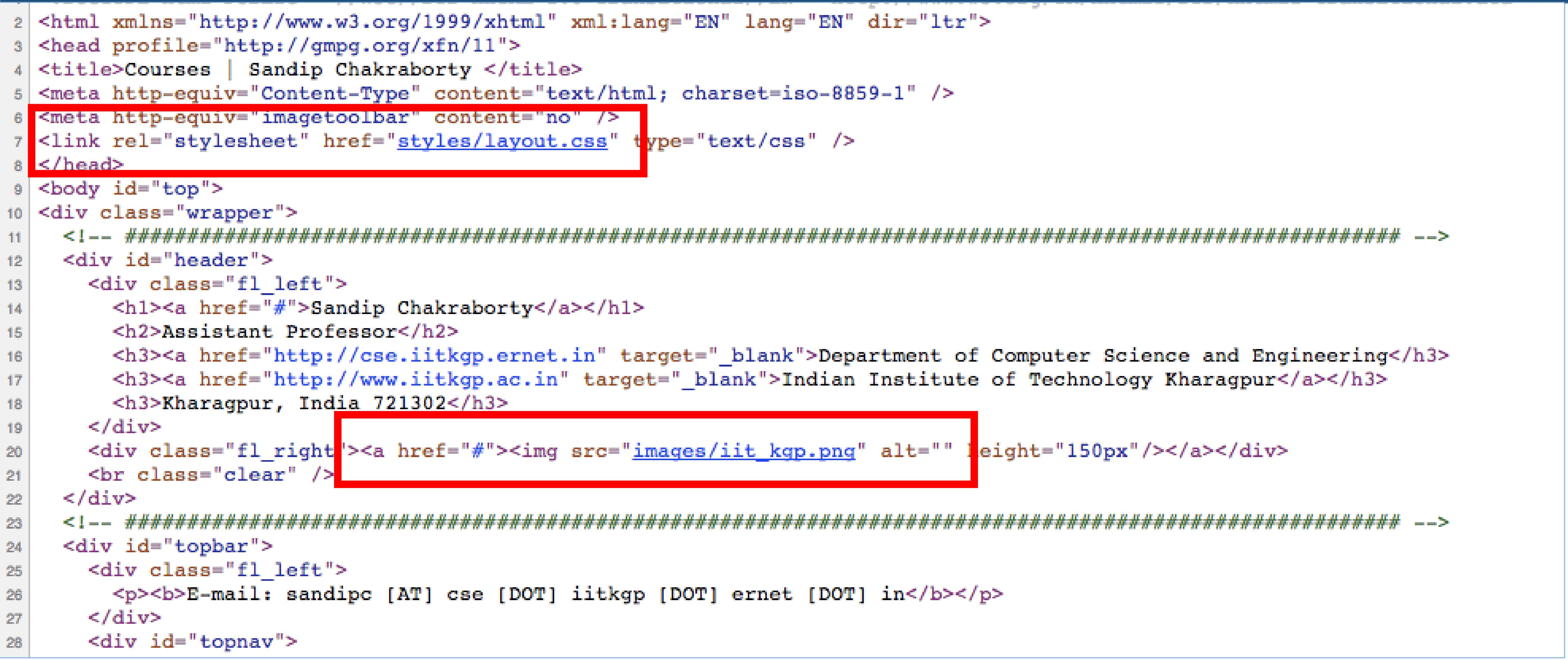
- **URL** have three components:
  - The protocol
  - The **qualified name** of the machine one which the page is located
  - The path uniquely indicating the specific page

http://cse.iitkgp.ac.in/~sandipc/courses.html

The Steps When You Click http://cse.iitkgp.ac.in/~sandipc/courses.html

- The browser determines the URL
- The browser asks DNS for the IP address of the server *cse.iitkgp.ac.in*
- DNS replies with 203.110.245.250
- The browser makes a TCP connection to 203.110.245.250 on port 80, the well known port for the HTTP protocol **(Note: https uses 443)**
- It sends over an HTTP request asking for the page *~sandipc/courses.html*
- The *cse.iitkgp.ac.in* server sends the page as an HTTP response, for example by sending the file */courses.html*
- If the page includes URLS that are needed for display, the browser fetches the other URLs using the same process

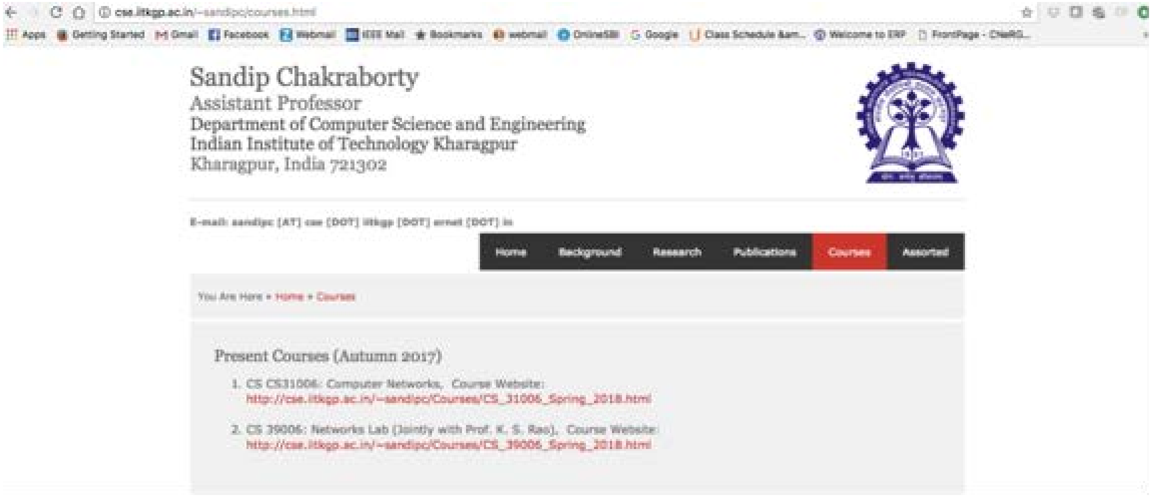
The Steps When You Click http://cse.iitkgp.ac.in/~sandipc/courses.html



```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="EN" lang="EN" dir="ltr">
<head profile="http://gmpg.org/xfn/11">
<title>Courses | Sandip Chakraborty </title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta http-equiv="imagetoolbar" content="no" />
<link rel="stylesheet" href="styles/layout.css" type="text/css" />
</head>
<body id="top">
<div class="wrapper">
  <!-- ####################################################################################################### -->
  <div id="header">
    <div class="fl_left">
      <h1><a href="#">Sandip Chakraborty</a></h1>
      <h2>Assistant Professor</h2>
      <h3><a href="http://cse.iitkgp.ernet.in" target="_blank">Department of Computer Science and Engineering</h3>
      <h3><a href="http://www.iitkgp.ac.in" target="_blank">Indian Institute of Technology Kharagpur</a></h3>
      <h3>Kharagpur, India 721302</h3>
    </div>
    <div class="fl_right"><a href="#"><img src="images/iit_kgp.png" alt="" height="150px"/></a></div>
    <br class="clear" />
  </div>
  <!-- ####################################################################################################### -->
  <div id="topbar">
    <div class="fl_left">
      <p><b>E-mail: sandipc [AT] cse [DOT] iitkgp [DOT] ernet [DOT] in</b></p>
    </div>
    <div id="topnav">
```

The Steps When You Click http://cse.iitkgp.ac.in/~sandipc/courses.html

- The browser displays the page *courses.html*
- The TCP connections are released if there are no other requests to the same server for a short period.



cse.iitkgp.ac.in/~sandipc/courses.html

Sandip Chakraborty
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, India 721302

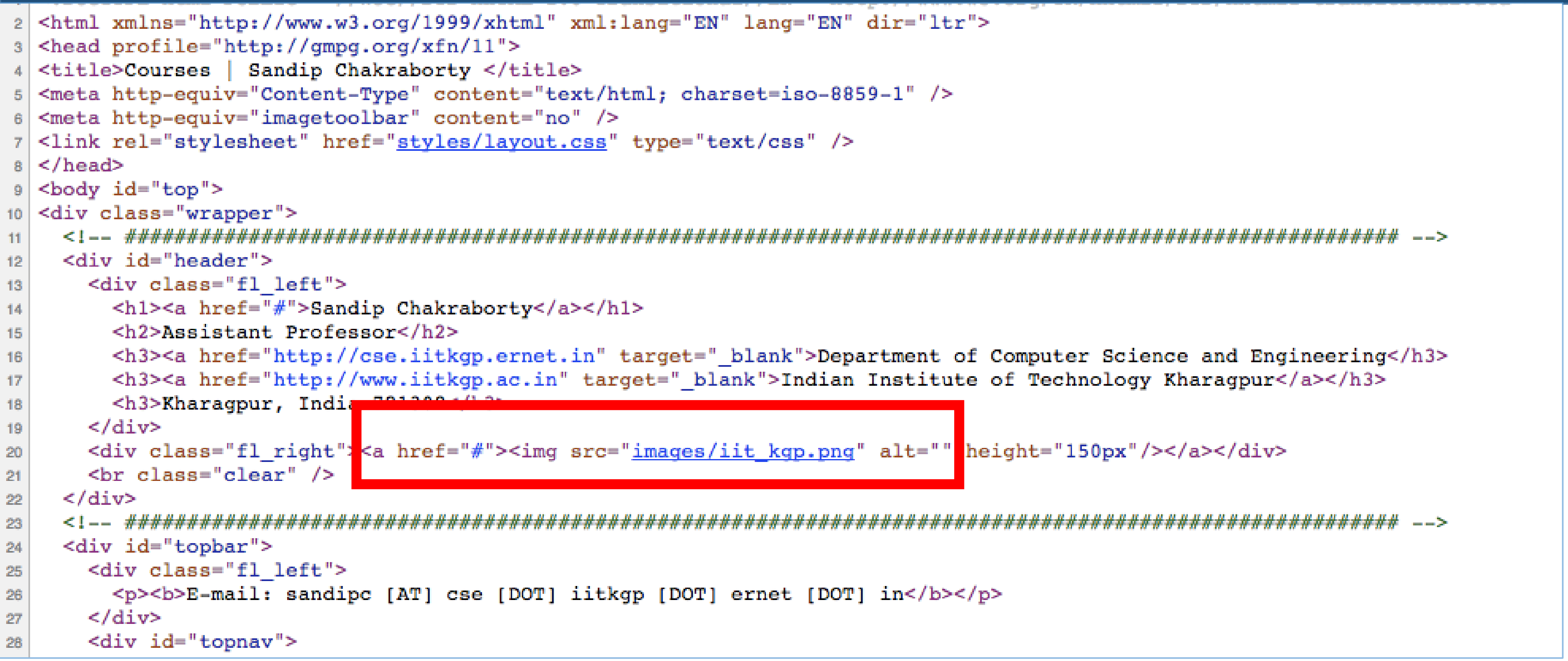
E-mail: sandipc [AT] cse [DOT] iitkgp [DOT] ernet [DOT] in

You Are Here » Home » Courses

Present Courses (Autumn 2017)

1. CS CS31006: Computer Networks, Course Website: http://cse.iitkgp.ac.in/~sandipc/Courses/CS_31006_Spring_2018.html
2. CS 39006: Networks Lab (Jointly with Prof. K. S. Rao), Course Website: http://cse.iitkgp.ac.in/~sandipc/Courses/CS_39006_Spring_2018.html

Uniform Resource Identifier (URI)



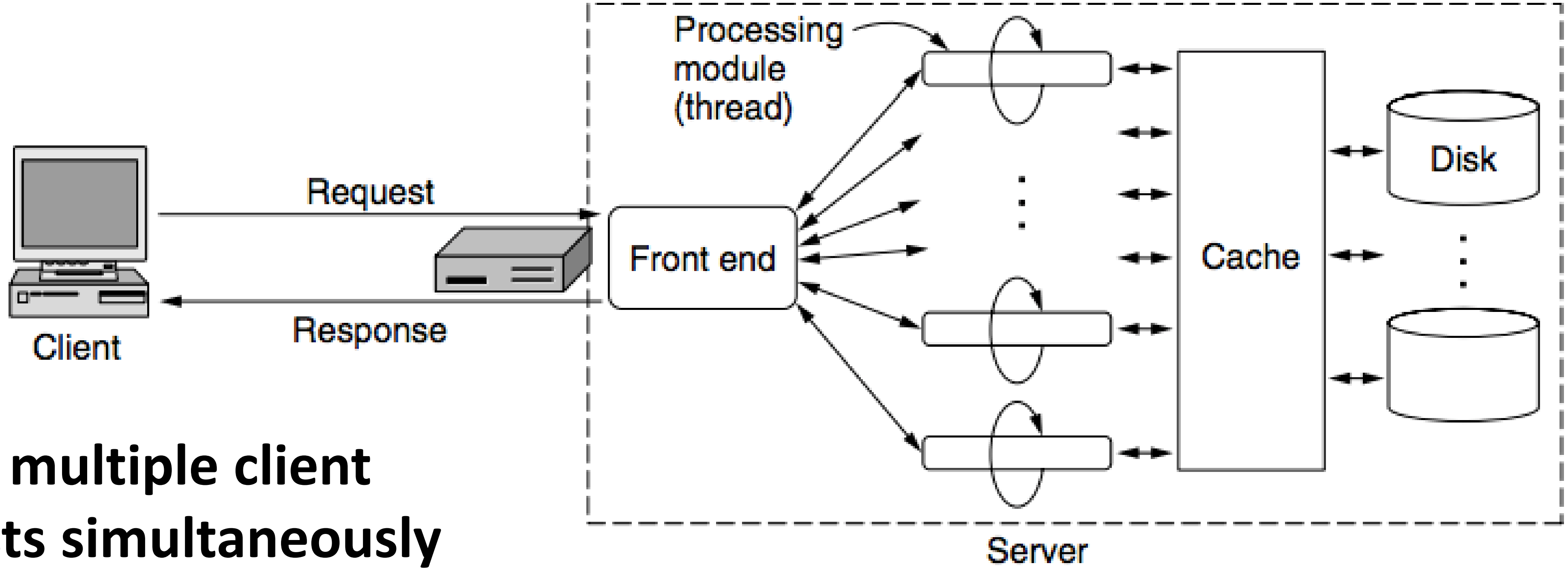
```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="EN" lang="EN" dir="ltr">
<head profile="http://gmpg.org/xfn/11">
<title>Courses | Sandip Chakraborty </title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta http-equiv="imagetoolbar" content="no" />
<link rel="stylesheet" href="styles/layout.css" type="text/css" />
</head>
<body id="top">
<div class="wrapper">
  <!-- ####################################################################################################### -->
  <div id="header">
    <div class="fl_left">
      <h1><a href="#">Sandip Chakraborty</a></h1>
      <h2>Assistant Professor</h2>
      <h3><a href="http://cse.iitkgp.ernet.in" target="_blank">Department of Computer Science and Engineering</h3>
      <h3><a href="http://www.iitkgp.ac.in" target="_blank">Indian Institute of Technology Kharagpur</a></h3>
      <h3>Kharagpur, India 721302</h3>
    </div>
    <div class="fl_right"><a href="#"><img src="images/iit_kgp.png" alt="" height="150px"/></a></div>
    <br class="clear" />
  </div>
  <!-- ####################################################################################################### -->
  <div id="topbar">
    <div class="fl_left">
      <p><b>E-mail: sandipc [AT] cse [DOT] iitkgp [DOT] ernet [DOT] in</b></p>
    </div>
    <div id="topnav">
```

- Generalization of the URLs – specifies the pages only or partially refers the pages without complete locations
- */images/iit_kgp.png* – may become URL *https://cse.iitkgp.ac.in/images/iit_kgp.png* if accessed from *cse.iitkgp.ac.in*

HTTP – The Server Side

- Accept a TCP connection from a client (a browser).
- Get the path to the page, which is the name of the file requested.
- Get the file (from disk).
- Sends the content of the file to the client.
- Release (close) the TCP connection.

Multi-Threaded Server



Serves multiple client requests simultaneously

Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

Connections

- HTTP uses TCP to set up a connection between the server and the client. In general, HTTP server runs at port 80 (default port) or 8080 (alternate port).

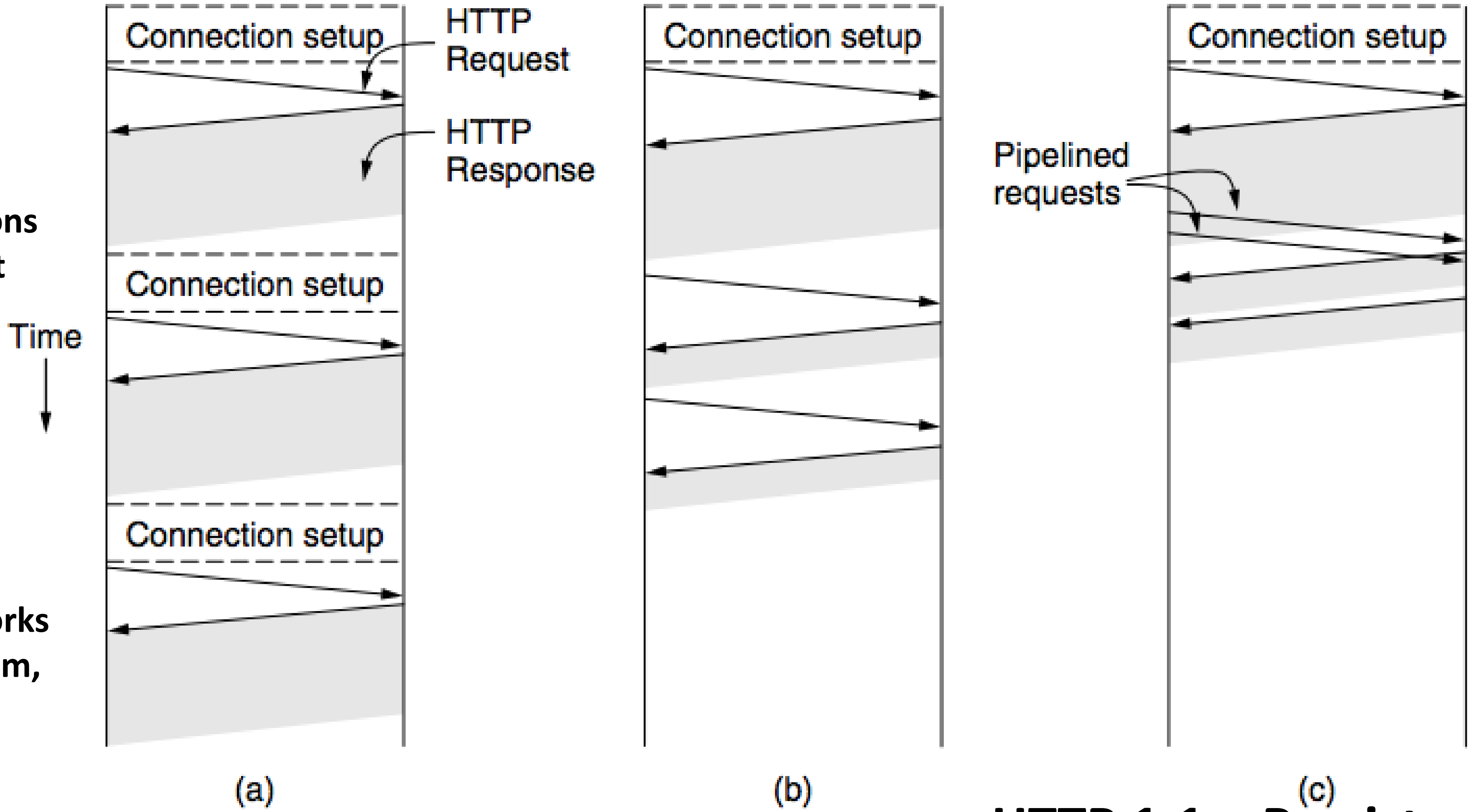
- HTTP 1.0 – After the connections were established, a single request was sent over and a single response was sent back. Then the TCP connections are released.
  - Create separate connections for every content in the web-page. Overhead is high.

- **Persistent Connection** (HTTP 1.1) – send additional requests and additional responses in a single TCP connection (**connection reuse)**.
  - It is also possible to **pipeline requests**.

Connections

Persistent Connections: Set at *keep-alive* information in the HTTP header (HTTP 1.0)

HTTP 1.1 – All connections are by default persistent



HTTP 1.0

HTTP 1.1 – Persistent Connections

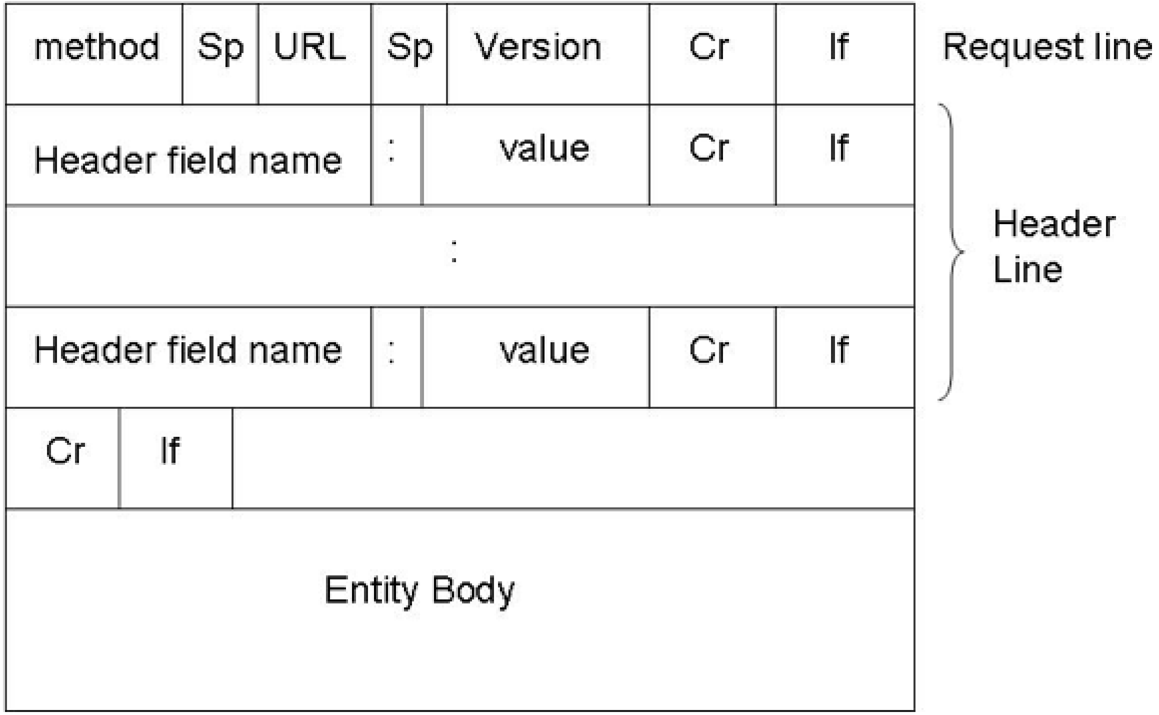
HTTP 1.1 – Persistent Connections (Pipelined)

Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

HTTP Request Methods

- Specifies what a HTTP Request will do

- GET *filename* HTTP/1.1



method | Sp | URL | Sp | Version | Cr | lf — Request line

Header field name | : | value | Cr | lf

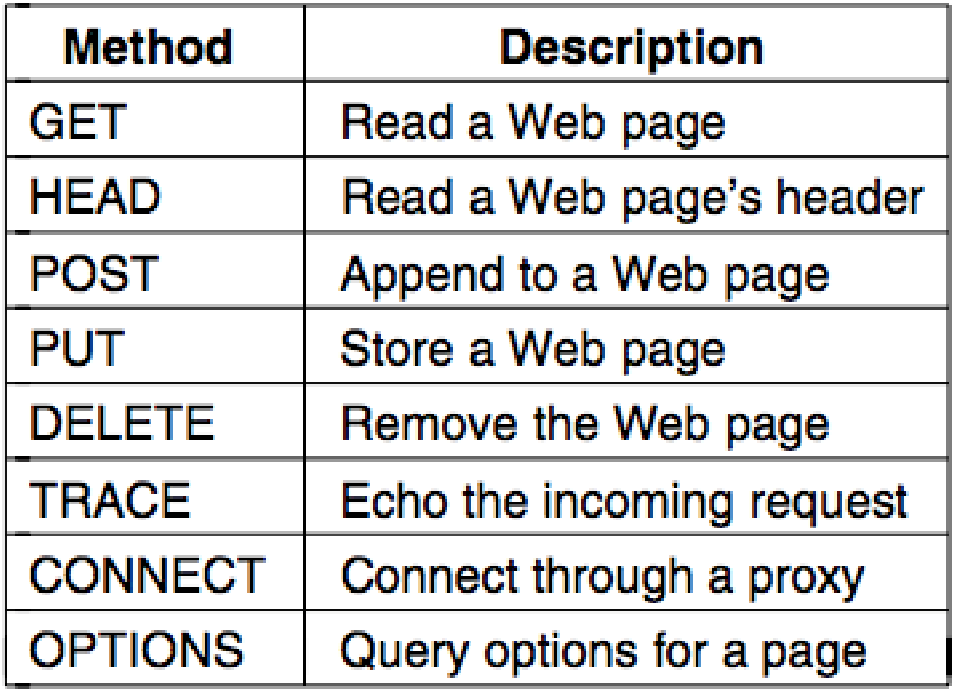
:

Header field name | : | value | Cr | lf

Header Line

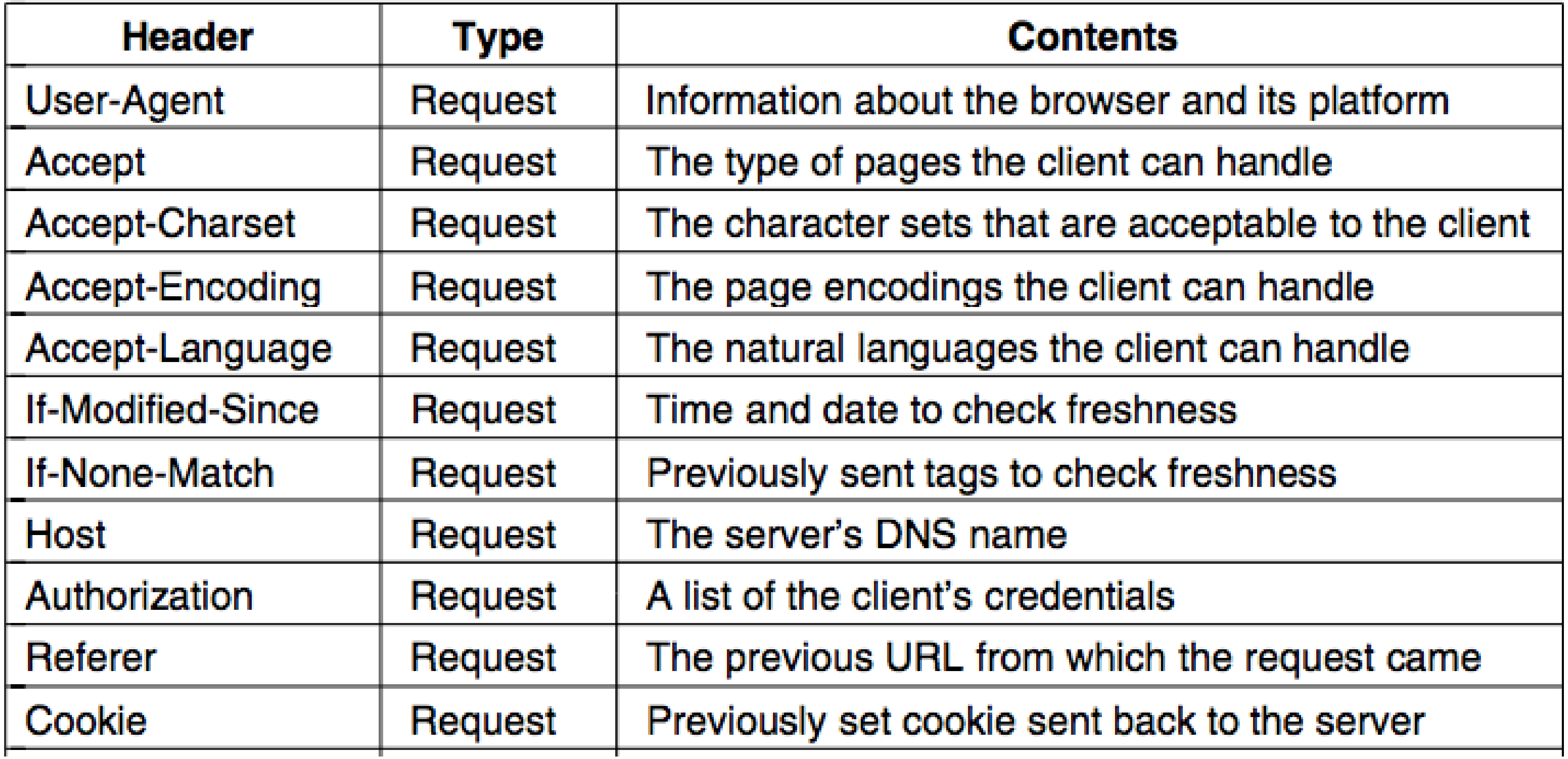
Cr | lf

Entity Body



Method	Description
GET	Read a Web page
HEAD	Read a Web page’s header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

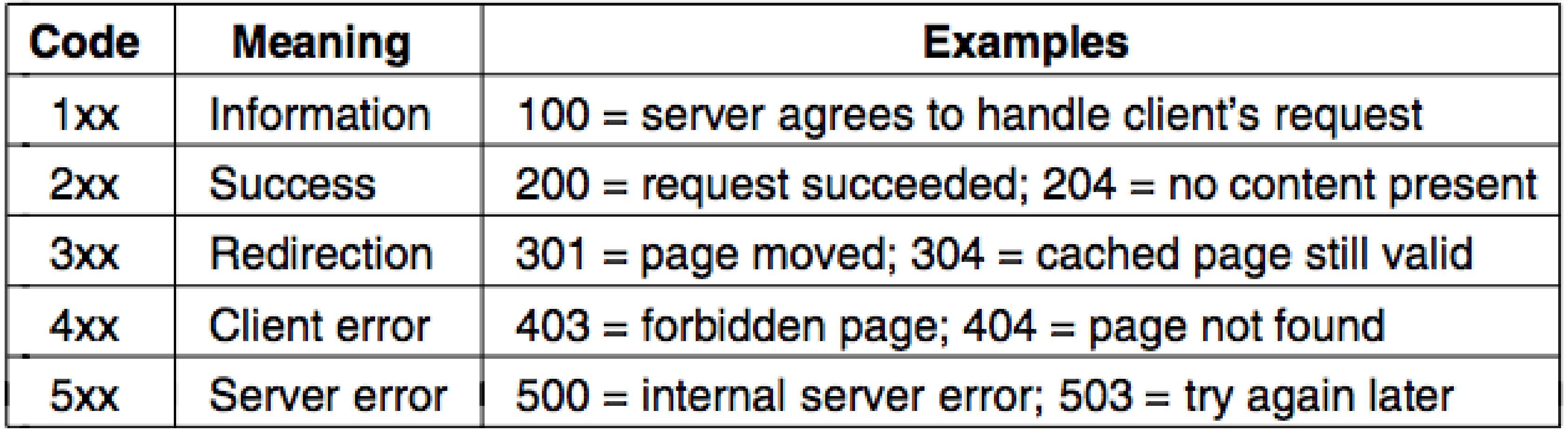
HTTP Request Header Fields (Partial List)



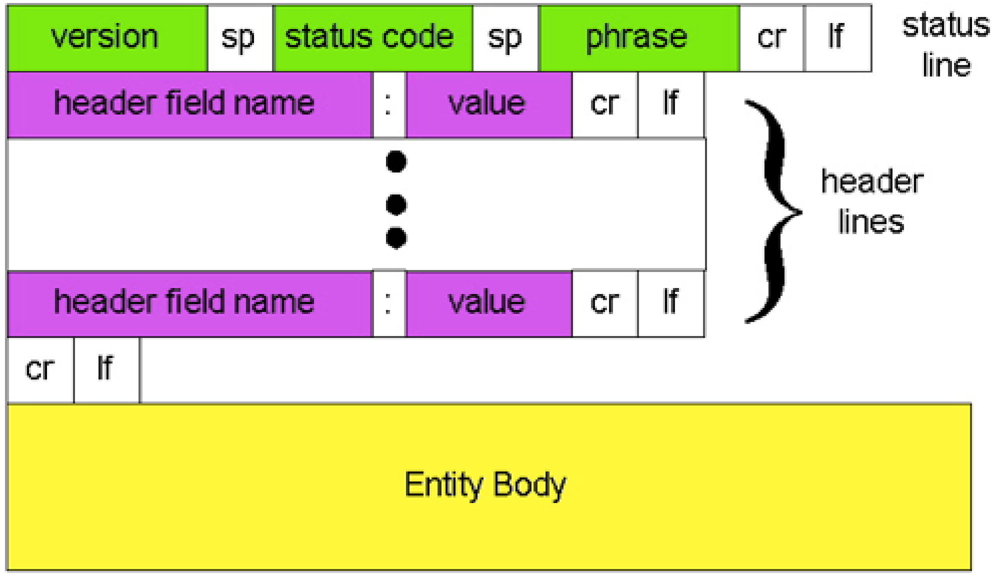
Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
If-Modified-Since	Request	Time and date to check freshness
If-None-Match	Request	Previously sent tags to check freshness
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Referer	Request	The previous URL from which the request came
Cookie	Request	Previously set cookie sent back to the server

HTTP Response

- Specifies the status of the request message.



Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later



version sp status code sp phrase cr lf — status line

header field name : value cr lf

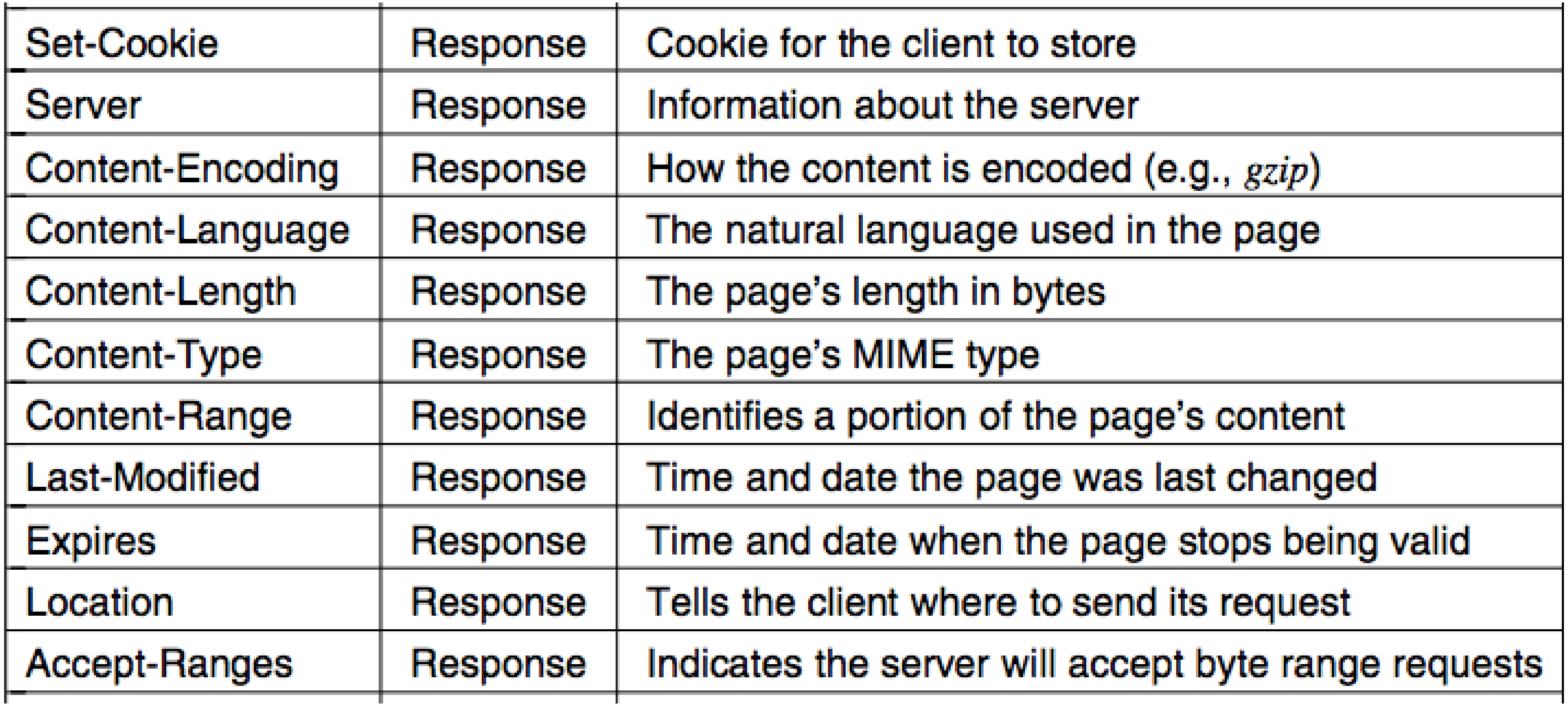
●
●
●

header field name : value cr lf — header lines

cr lf

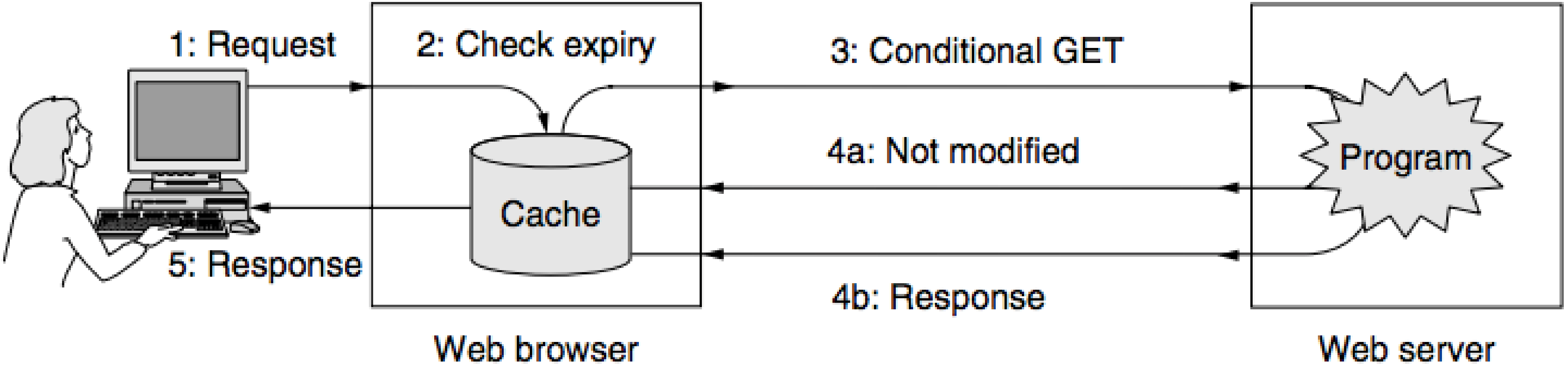
Entity Body

HTTP Response Header Fields (Partial List)



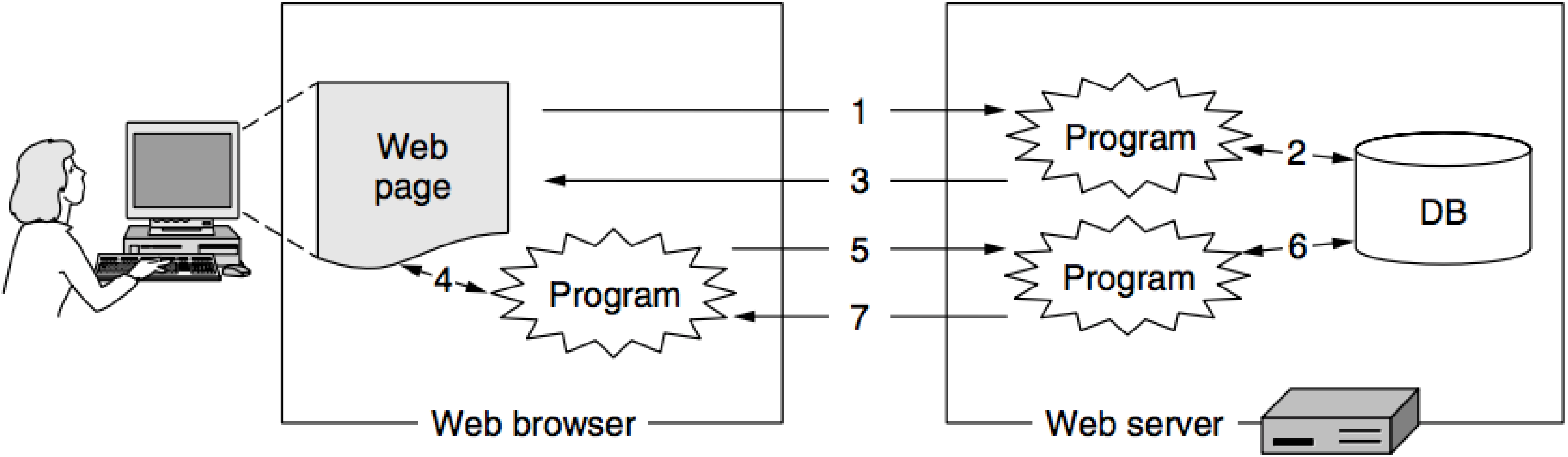
Set-Cookie	Response	Cookie for the client to store
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., *gzip*)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Content-Range	Response	Identifies a portion of the page's content
Last-Modified	Response	Time and date the page was last changed
Expires	Response	Time and date when the page stops being valid
Location	Response	Tells the client where to send its request
Accept-Ranges	Response	Indicates the server will accept byte range requests

HTTP Caching



Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

Dynamic Web Applications

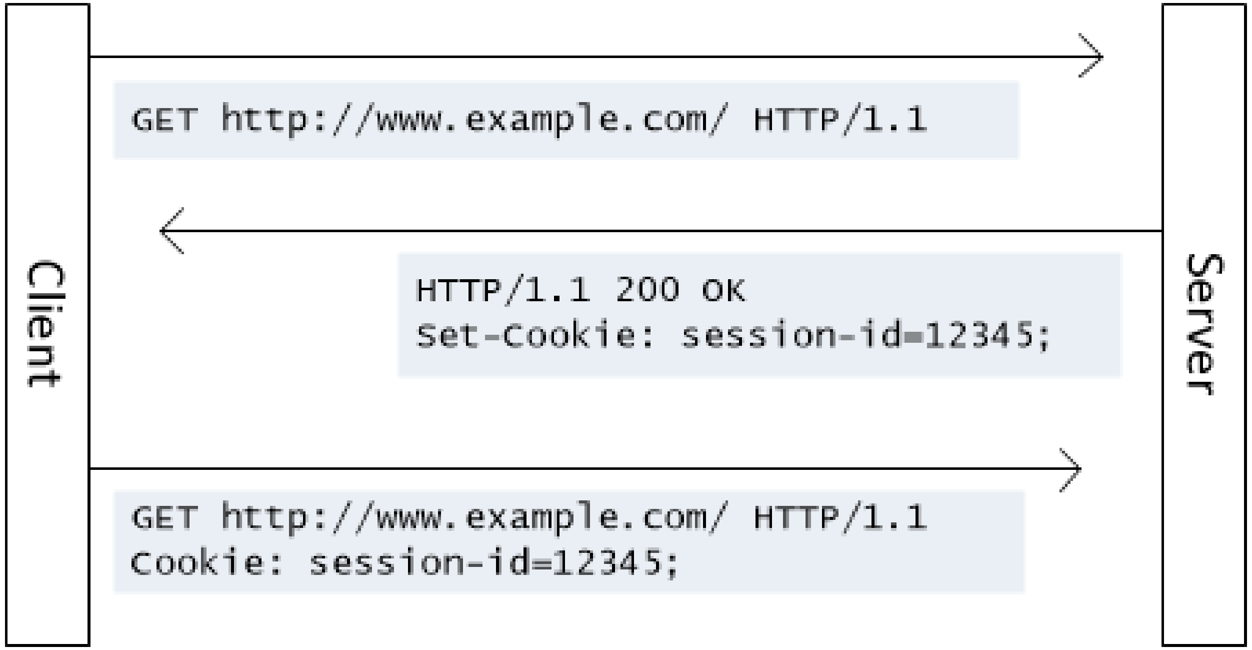


Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

Cookies

- HTTP is by default a **stateless protocol**
  - Every Response corresponds to the previous Request only, it does not remember any state information, such as last page accessed

- Use **Cookies** to store the state information. Client forwards the additional information along with the Request message by reading the cookie.



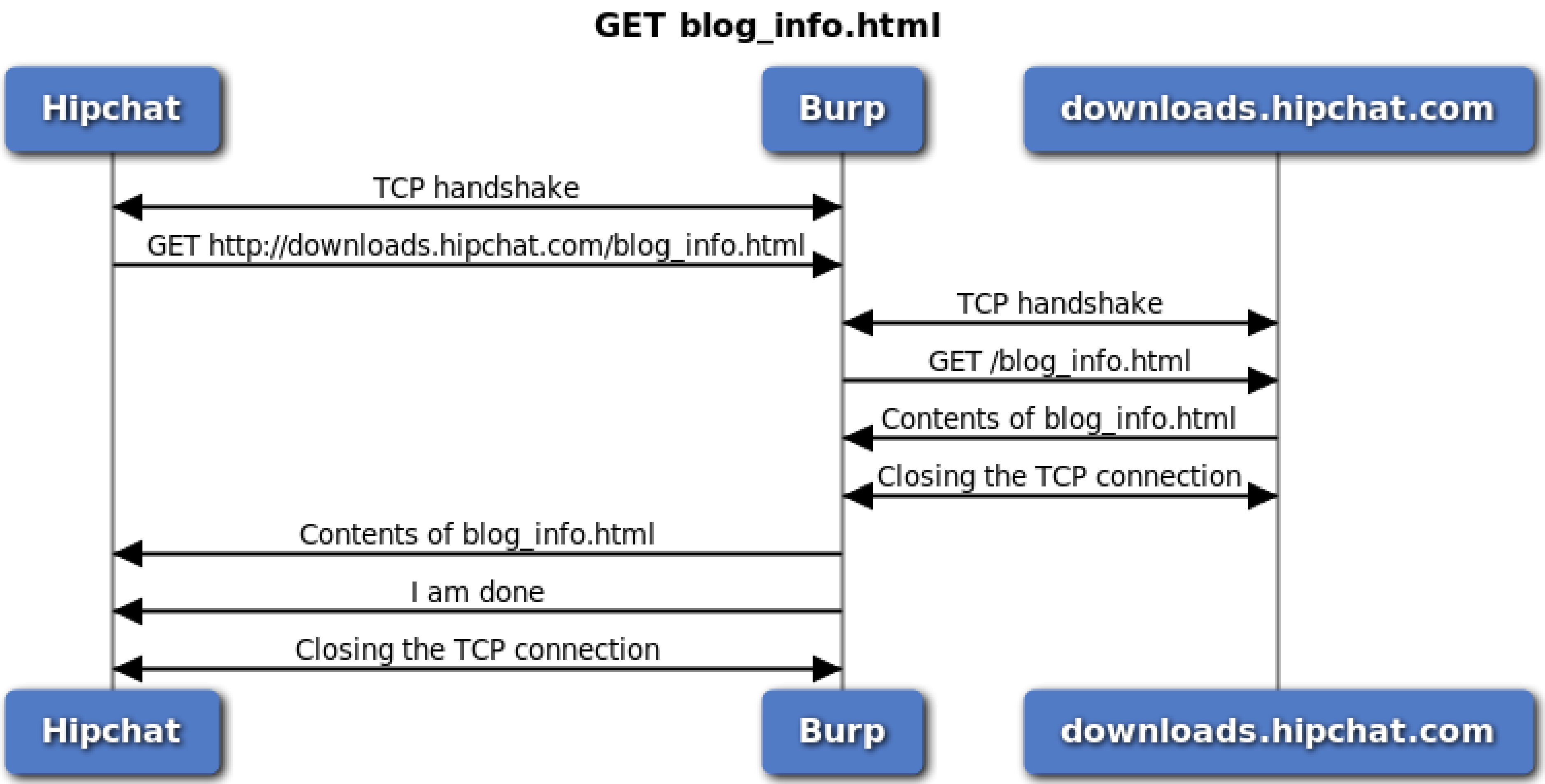
```
GET http://www.example.com/ HTTP/1.1
```

```
HTTP/1.1 200 OK
Set-Cookie: session-id=12345;
```

```
GET http://www.example.com/ HTTP/1.1
Cookie: session-id=12345;
```

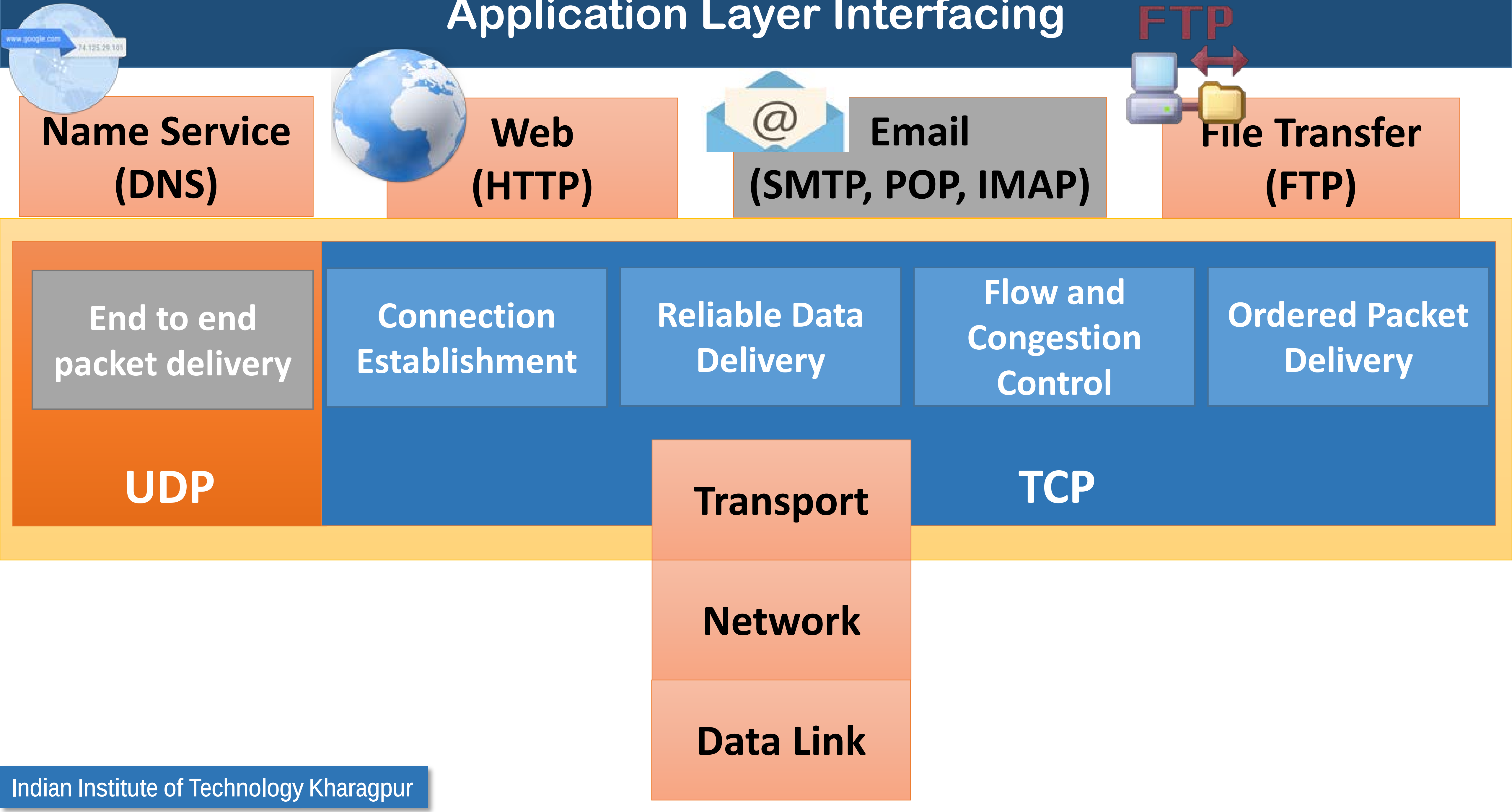
HTTP Proxy

- Intercepts the TCP connections to process the HTTP data.


GET blog_info.html
Hipchat
Burp
downloads.hipchat.com
TCP handshake
GET http://downloads.hipchat.com/blog_info.html
TCP handshake
GET /blog_info.html
Contents of blog_info.html
Closing the TCP connection
Contents of blog_info.html
I am done
Closing the TCP connection
Hipchat
Burp
downloads.hipchat.com

HTTP Proxy

Application Layer Interfacing



Name Service (DNS)

Web (HTTP)

Email (SMTP, POP, IMAP)

File Transfer (FTP)

End to end packet delivery

UDP

Connection Establishment

Reliable Data Delivery

Flow and Congestion Control

Ordered Packet Delivery

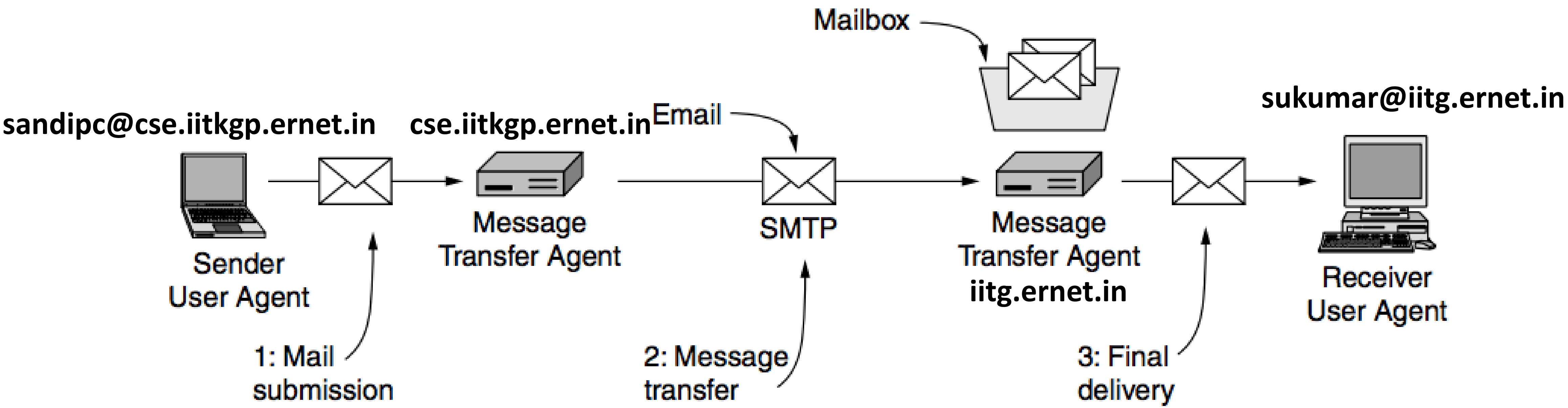
TCP

Transport

Network

Data Link

Electronic Mails – Architecture and Services



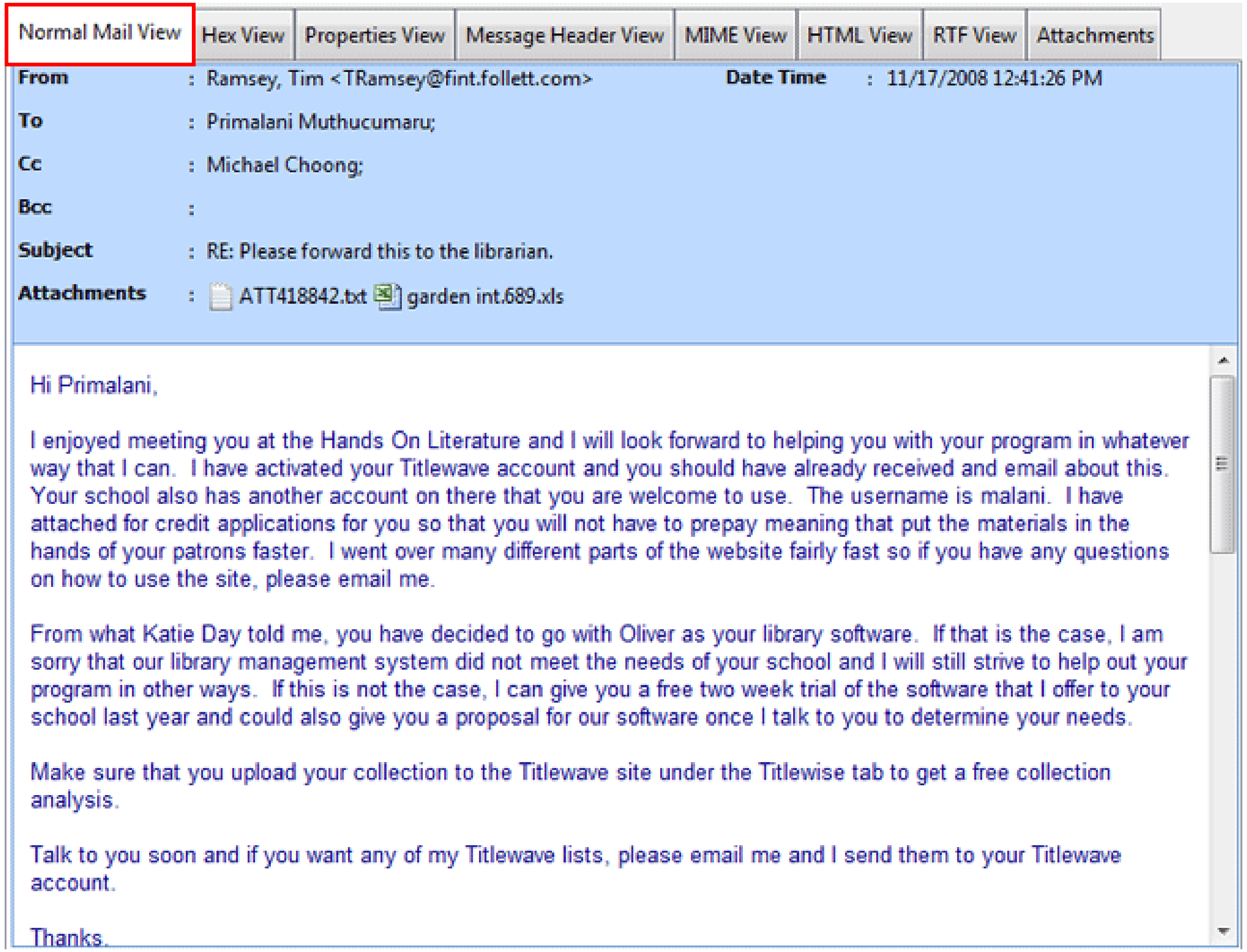
- **User Agent:** Allow people to read and send emails.
- **Message Transfer Agents (main servers):** Move the message from the source to the destination

Message Transfer Agents

- System processes run in the background on mail servers (always available).
- Automatically move emails through the system from the originator to the recipient
- Uses **Simple Mail Transfer Protocol (SMTP)** – RFC 821, RFC 5321
- Implements **mailing lists,** an identical copy of message is delivered to everyone in the list (btech@iitkgp.ac.in)
- Implements **Mailboxes**, to store all the emails received for a user

Message Format (RFC 5322)

- An envelope containing message header and message body



From : Ramsey, Tim <TRamsey@fint.follett.com> **Date Time** : 11/17/2008 12:41:26 PM

To : Primalani Muthucumaru;

Cc : Michael Choong;

Bcc :

Subject : RE: Please forward this to the librarian.

Attachments : ATT418842.txt garden int.689.xls

Hi Primalani,

I enjoyed meeting you at the Hands On Literature and I will look forward to helping you with your program in whatever way that I can. I have activated your Titlewave account and you should have already received and email about this. Your school also has another account on there that you are welcome to use. The username is malani. I have attached for credit applications for you so that you will not have to prepay meaning that put the materials in the hands of your patrons faster. I went over many different parts of the website fairly fast so if you have any questions on how to use the site, please email me.

From what Katie Day told me, you have decided to go with Oliver as your library software. If that is the case, I am sorry that our library management system did not meet the needs of your school and I will still strive to help out your program in other ways. If this is not the case, I can give you a free two week trial of the software that I offer to your school last year and could also give you a proposal for our software once I talk to you to determine your needs.

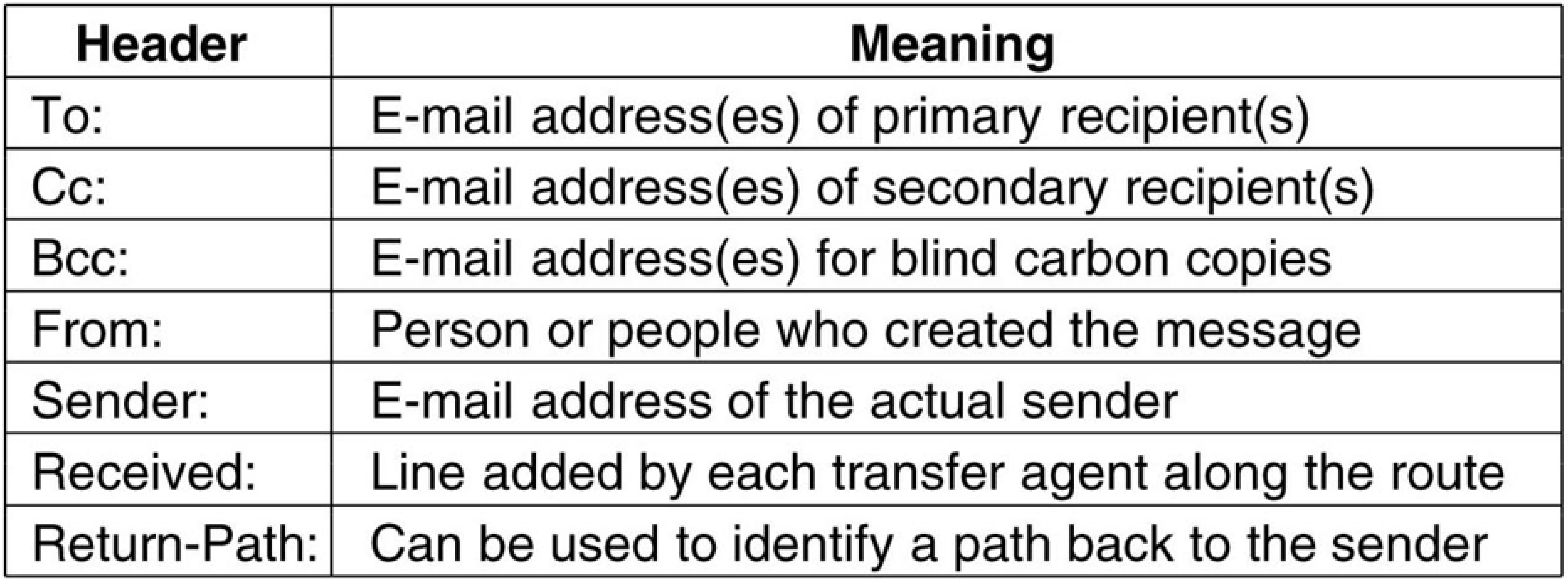
Make sure that you upload your collection to the Titlewave site under the Titlewise tab to get a free collection analysis.

Talk to you soon and if you want any of my Titlewave lists, please email me and I send them to your Titlewave account.

Thanks.

The Internet Message Format (RFC 5322)

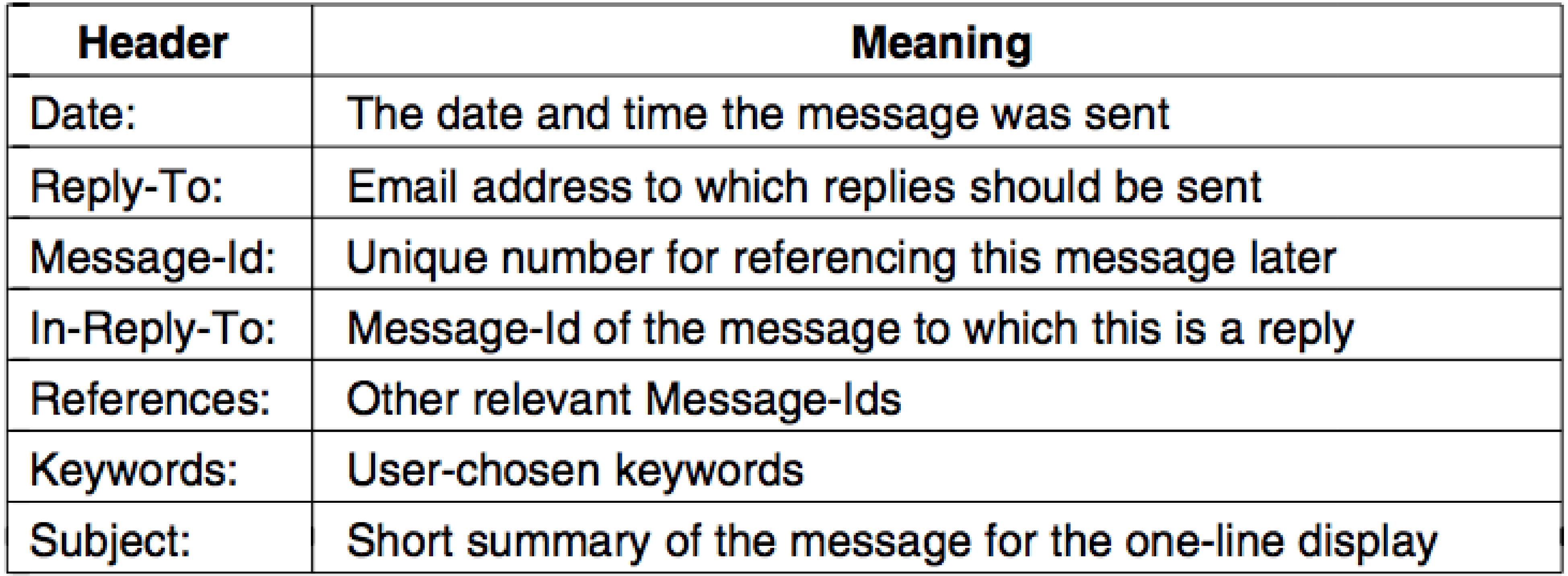
- **Header fields (for message transport):**



Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

The Internet Message Format (RFC 5322)

- **Header fields (additional fields for message description):**



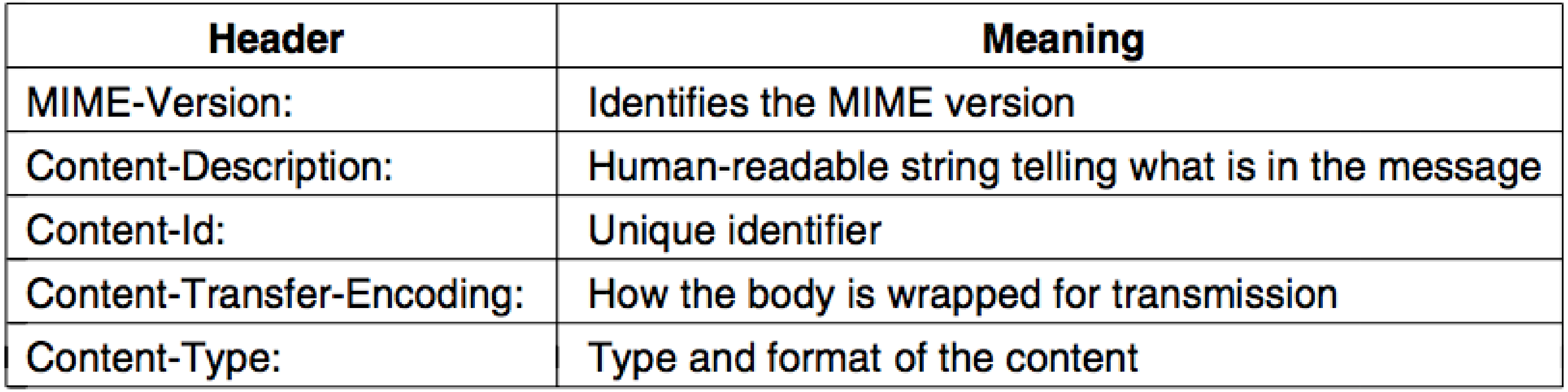
Header	Meaning
Date:	The date and time the message was sent
Reply-To:	Email address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display

MIME – The Multipurpose Internet Mail Extension

- ARPANET: email consisted exclusively of text messages written in English and expressed in ASCII

- MIME: Use multi-language and multimedia contents (audio, image etc.) inside an email.

- **Additional message headers for MIME:**

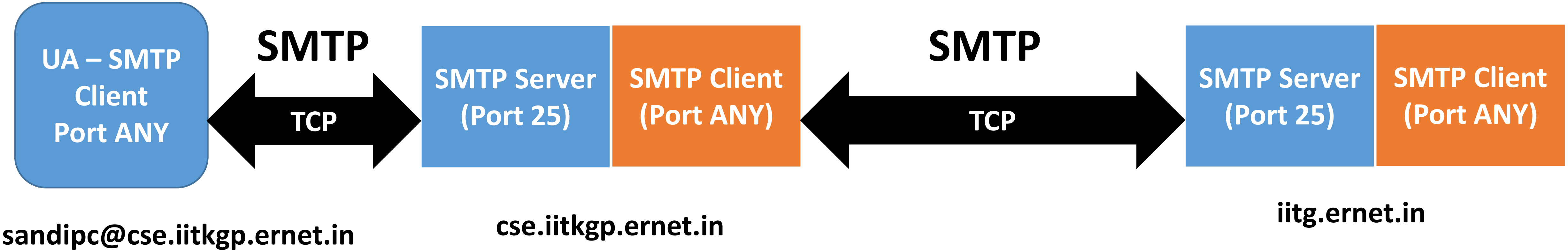


Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

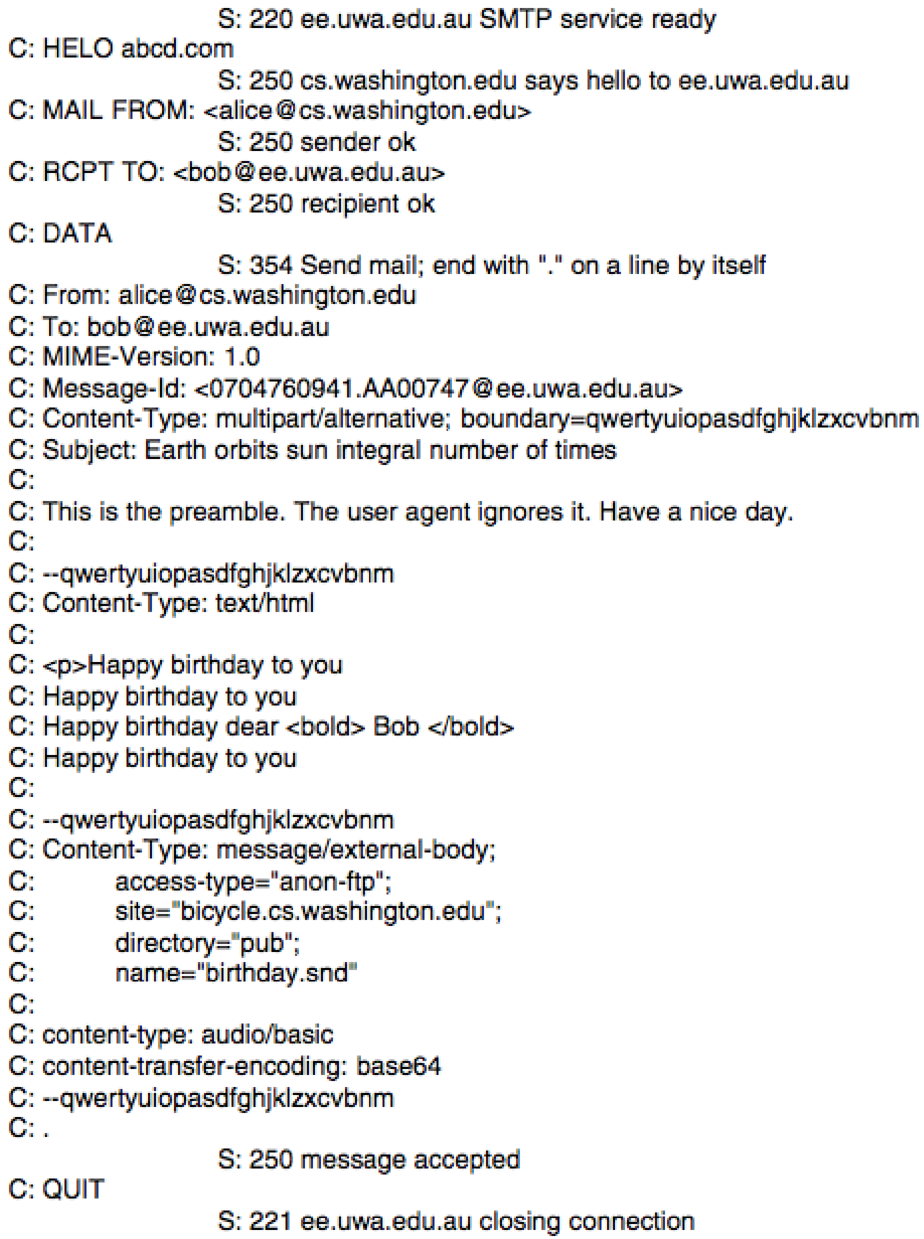
Message Transfer

- Uses SMTP Protocol

- Email is delivered by having the sending computer establishing TCP connection to port 25 of the receiving computer.



Message Transfer (SMTP)



```
                        S: 220 ee.uwa.edu.au SMTP service ready
C: HELO abcd.com
                        S: 250 cs.washington.edu says hello to ee.uwa.edu.au
C: MAIL FROM: <alice@cs.washington.edu>
                        S: 250 sender ok
C: RCPT TO: <bob@ee.uwa.edu.au>
                        S: 250 recipient ok
C: DATA
                        S: 354 Send mail; end with "." on a line by itself
C: From: alice@cs.washington.edu
C: To: bob@ee.uwa.edu.au
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@ee.uwa.edu.au>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/html
C:
C: <p>Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Bob </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C:          access-type="anon-ftp";
C:          site="bicycle.cs.washington.edu";
C:          directory="pub";
C:          name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
                        S: 250 message accepted
C: QUIT
                        S: 221 ee.uwa.edu.au closing connection
```

Source: Computer Networks (5th Edition) by Tanenbaum, Wetherell

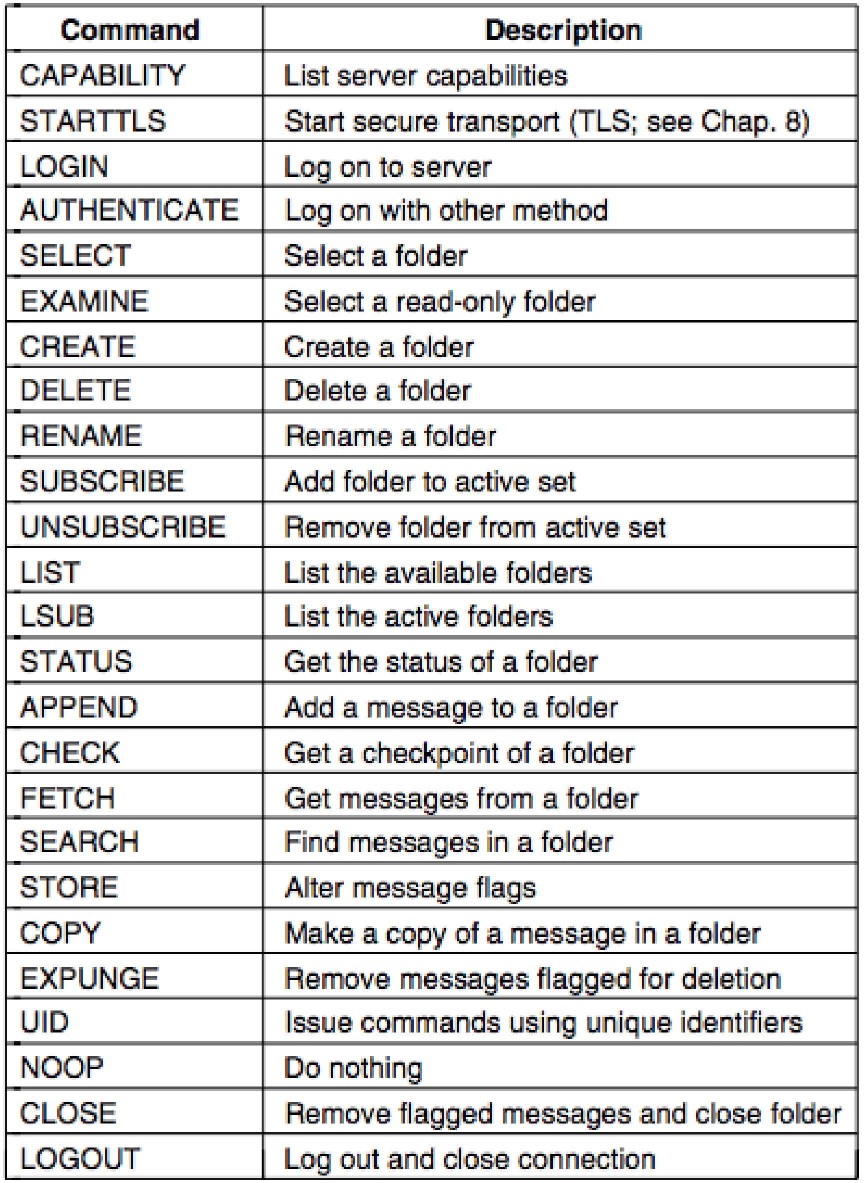
Final Delivery

- Pull type protocol – UA at the receiver side pulls the emails from mail server after login.

- **Post Office Protocol, Version 3** (POP3) – an earlier protocol for email delivery

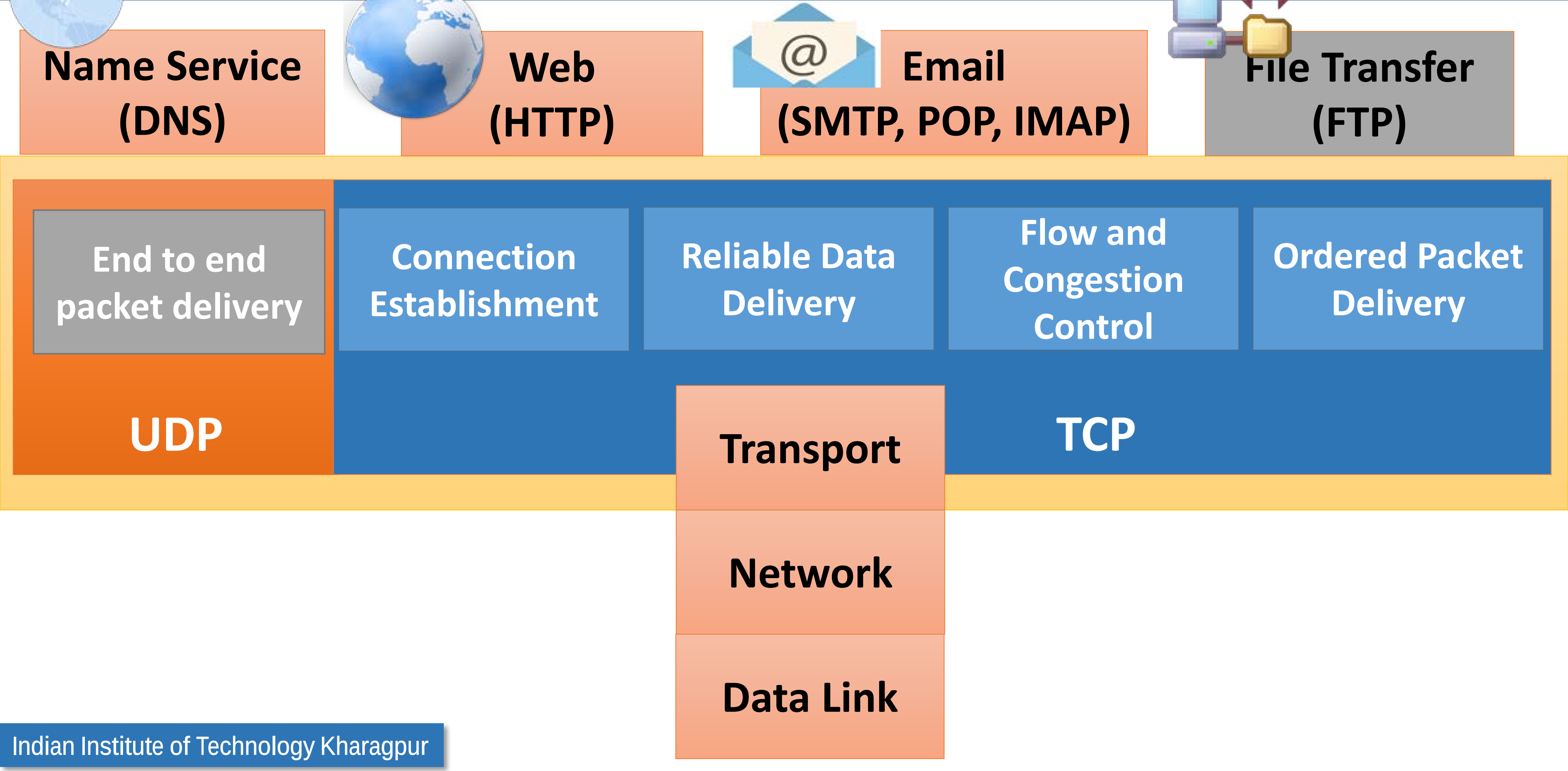
- **Internet Message Access Protocol, Version 4** (IMAP v4) – RFC 3501
  - The email server runs an IMAP server at port 143
  - The user agent runs IMAP client
  - The client connects to the server and issues mail delivery commands

IMAP (Version 4) Commands



Command	Description
CAPABILITY	List server capabilities
STARTTLS	Start secure transport (TLS; see Chap. 8)
LOGIN	Log on to server
AUTHENTICATE	Log on with other method
SELECT	Select a folder
EXAMINE	Select a read-only folder
CREATE	Create a folder
DELETE	Delete a folder
RENAME	Rename a folder
SUBSCRIBE	Add folder to active set
UNSUBSCRIBE	Remove folder from active set
LIST	List the available folders
LSUB	List the active folders
STATUS	Get the status of a folder
APPEND	Add a message to a folder
CHECK	Get a checkpoint of a folder
FETCH	Get messages from a folder
SEARCH	Find messages in a folder
STORE	Alter message flags
COPY	Make a copy of a message in a folder
EXPUNGE	Remove messages flagged for deletion
UID	Issue commands using unique identifiers
NOOP	Do nothing
CLOSE	Remove flagged messages and close folder
LOGOUT	Log out and close connection

Application Layer Interfacing



Name Service (DNS)

Web (HTTP)

Email (SMTP, POP, IMAP)

File Transfer (FTP)

End to end packet delivery

UDP

Connection Establishment

Reliable Data Delivery

Flow and Congestion Control

Ordered Packet Delivery

TCP

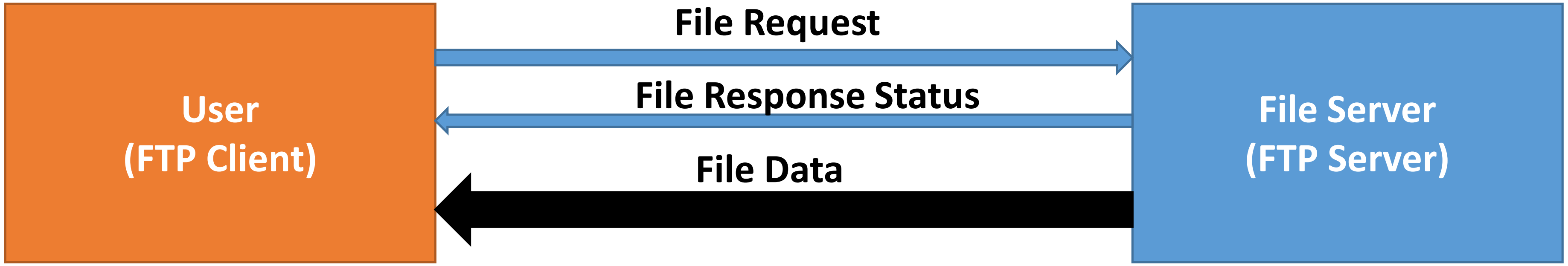
Transport

Network

Data Link

File Transfer Protocol (FTP)

- Is built on a client-server model (RFC 959)
  - The client requests for the file or send the file to the server
  - The server responses with the file data or store the file at the file server



- Works in two modes – **Active and Passive**

Active and Passive Modes of File Transfer

FTP server uses two different ports:

Port 21 (Command or Control Port): For command message transfer

Port 20 or Client assigned (Data Port): For data transfer

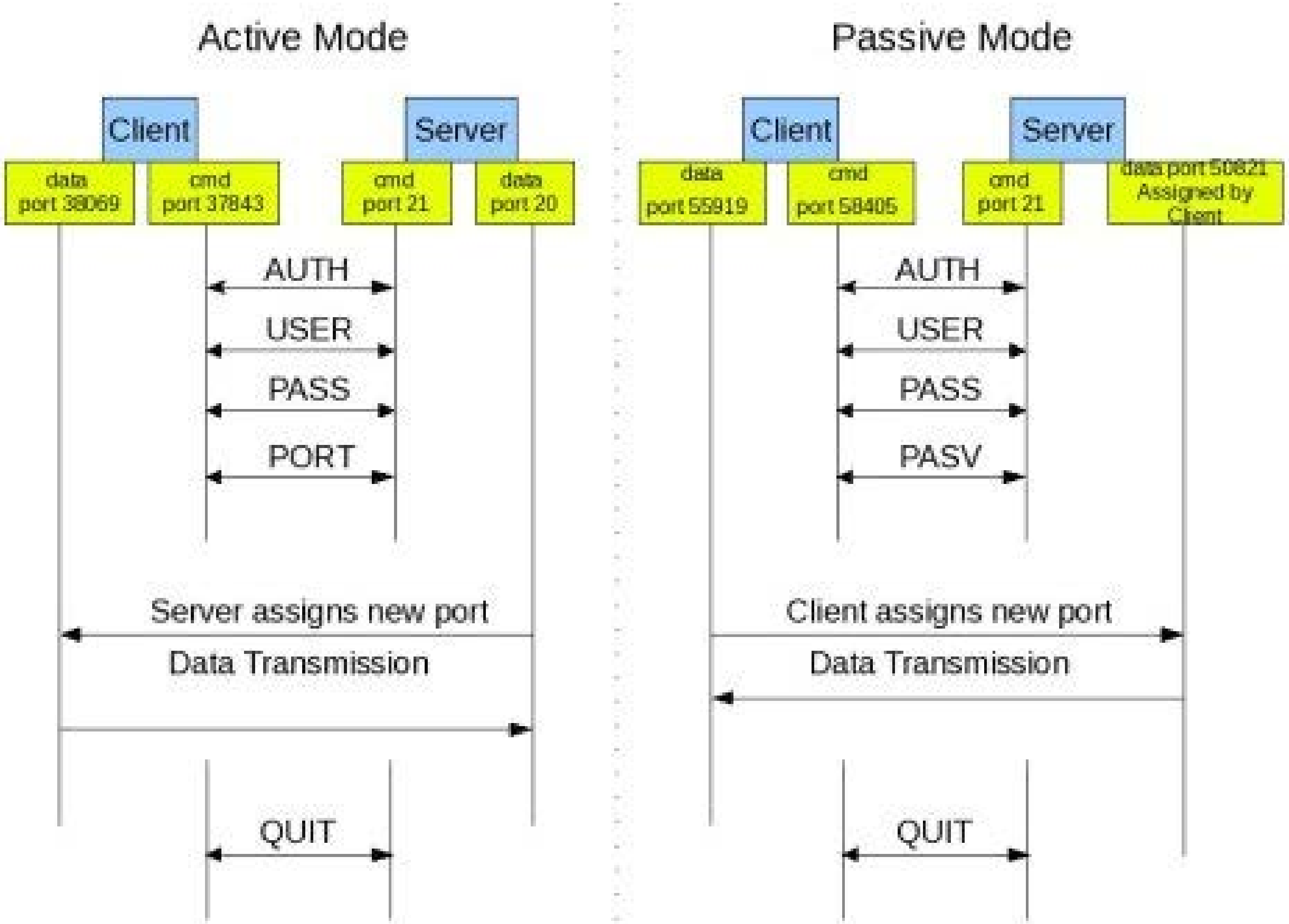


Image Source: http://henrydu.com/blog/how-to/ftp-active-mode-vs-passive-mode-106.html

Why There are Two Channels – Command Channel and Data Channel

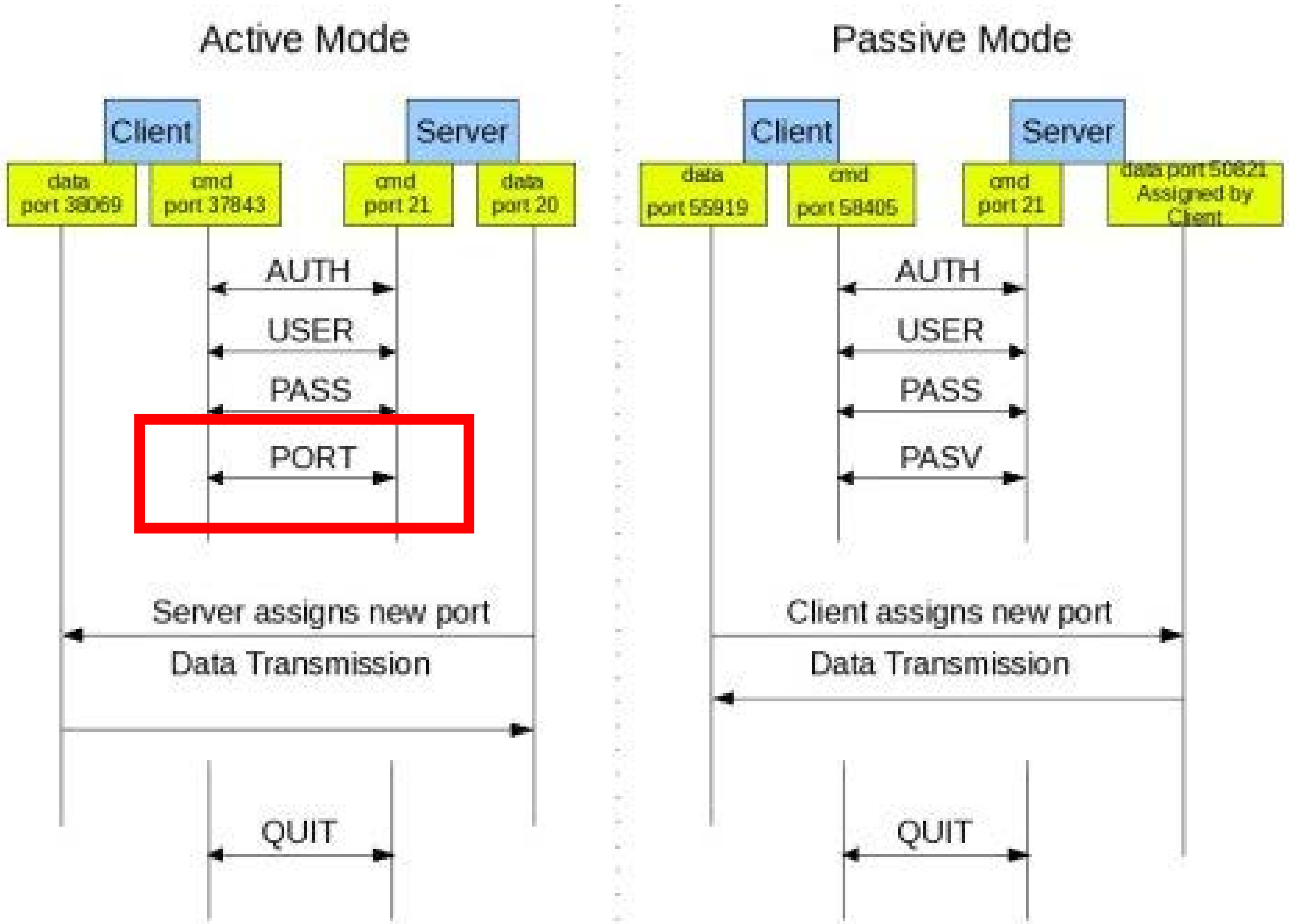
- Specifically to avoid busy waiting, and keep the command channel lightweight.

- **You can always use a multiplexing between command/control and data,** **but** FTP is used for large file transfer; if command channel is used for data transfer as well, the commands for other clients may experience a higher queuing delay while one client is being served.

- The clients can continue sending and receiving control information while data transfer is being take place

Why There are Two Modes in FTP?

Active Mode: Client informs the port number where it is listening, and the server initiates the TCP connection to that port (TCP server is running at the client side)



What If the client is behind a firewall and can not accept a connection?

Why There are Two Modes in FTP?

Passive Mode: The server selects a random port, and the client initiates a TCP connection to that server port.

The server can serve multiple clients at different server data ports through different threads.

The clients always initiate the command and the data transfer.

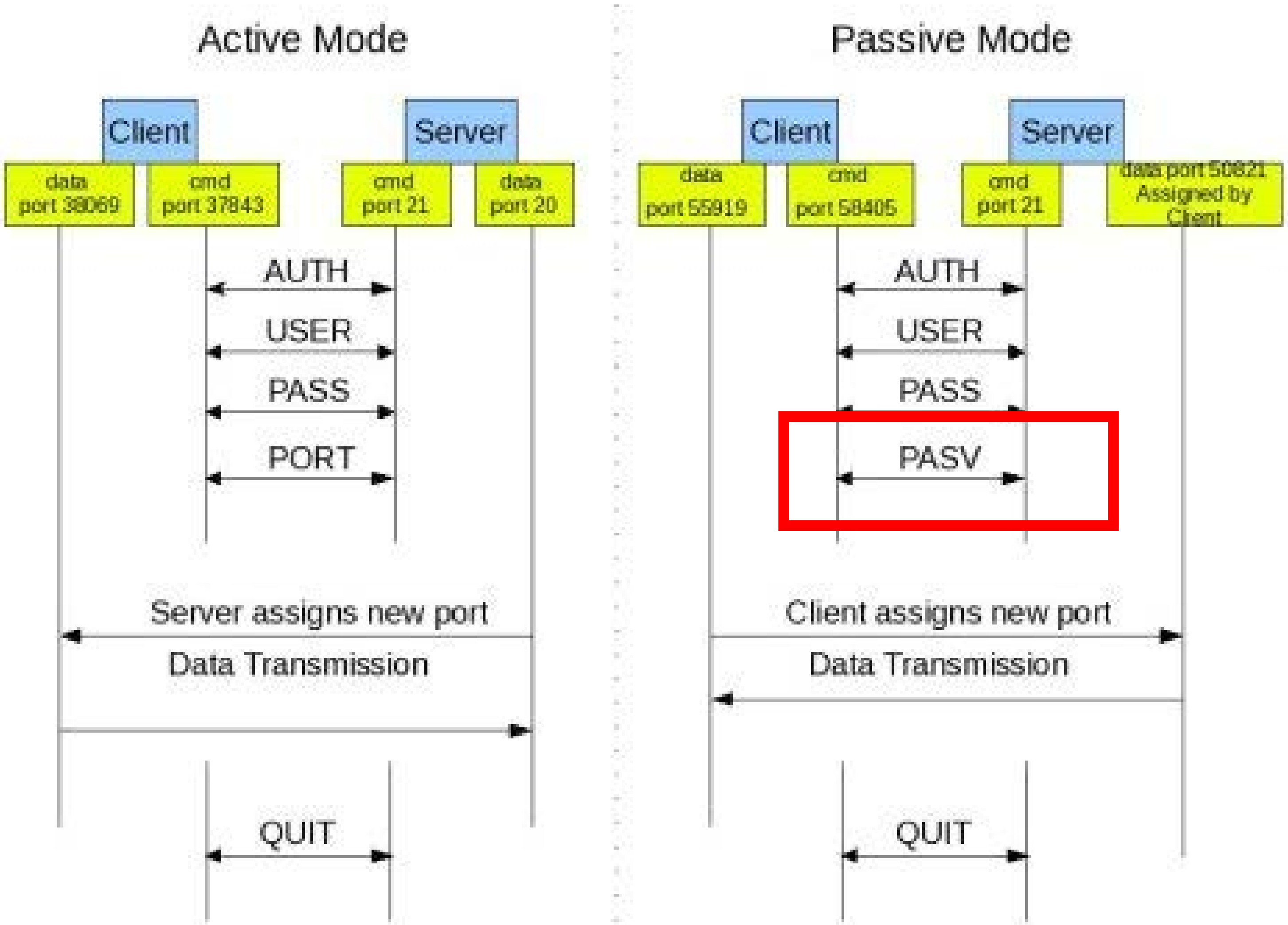


Image Source: http://henrydu.com/blog/how-to/ftp-active-mode-vs-passive-mode-106.html

FTP Data Transfer Modes

- **Stream mode**: Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.

- **Block mode:** FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.

- **Compressed mode**: Data is compressed using a simple algorithm (usually run-length encoding).

Source: Wikipedia

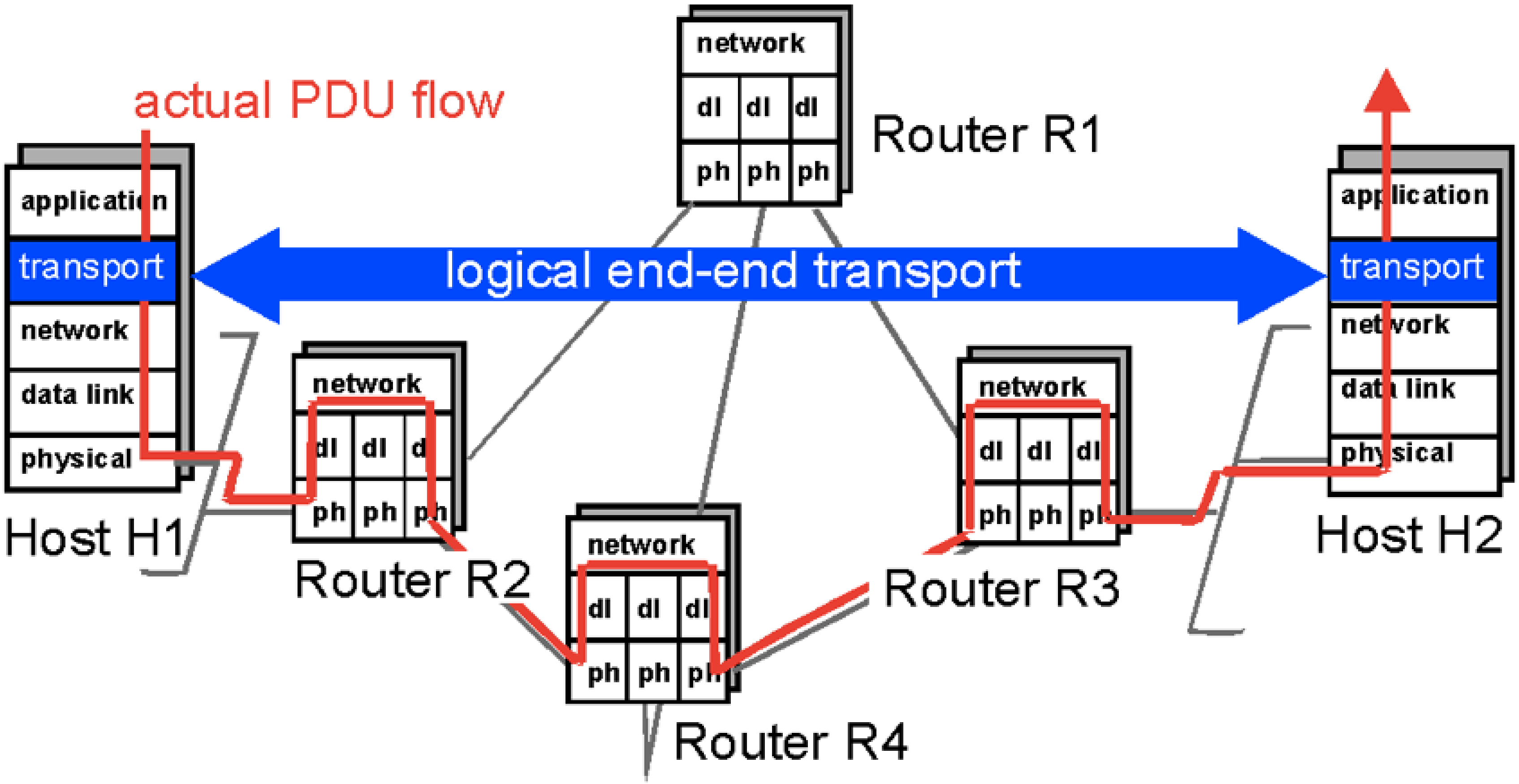
FTP Sample Commands and Response Codes

Sample commands:

- Sent as ASCII text over control channel
- `USER` *`username`*
- `PASS` *`password`*
- `LIST`: Return list of file in current directory
- `RETR filename`: Retrieves (gets) file
- `STOR filename`: Stores (puts) file onto remote host

Sample return codes

- Status code and phrase (as in HTTP)
- `331 username OK, password required`
- `125 data connection already open; transfer starting`
- `425 can’t open data connection`
- `452 error writing file`



- Next, we'll go for the Transport Layer ....